

Návrh a optimalizace dodavatelských systémů s využitím dynamické simulace

Jakub Dyntar

OBSAH

1	ÚVOD	5
2	CÍLE A STRUKTURA MONOGRAFIE	6
3	SOUČASNÉ POJETÍ KONCEPCE LOGISTIKY	7
3.1	Koncept supply chain.....	7
3.2	Supply chain management	9
4	SOUČASNÝ STAV MODELOVÁNÍ DODAVATELSKÝCH SYSTÉMŮ	12
4.1	Kvantitativní přístupy používané při modelování dodavatelských systémů	12
4.1.1	Matematické programování	14
4.1.1.1	Lineární programování	14
4.1.1.2	Celočíselné a smíšené programování	15
4.1.1.3	Nelineární programování	15
4.1.1.4	Vícekriteriální programování	16
4.1.1.5	Dynamické programování	17
4.1.1.6	Teorie front	21
4.1.2	Heuristické metody.....	24
4.1.2.1	Petriho síť	25
4.1.2.2	Neuronová síť	27
4.1.2.3	Fuzzy logic.....	30
4.1.2.4	Case based reasoning	31
4.1.2.5	Rough set	32
4.1.2.6	Genetický algoritmus	33
4.1.2.7	Ant colony optimalizace	35
4.1.2.8	Particle swarm optimalizace	36
4.1.2.9	Simulované žíhání.....	36
4.1.2.10	Tabu search	39
4.1.3	Analytické modely.....	41
4.1.3.1	Teorie her.....	41
4.1.3.2	Data envelopment analysis	42
4.1.3.3	Analytic hierarchy process.....	44
4.1.3.4	Analytic network process.....	45
4.1.3.5	Life cycle analysis	45
4.2	Supply chain operations reference model	47
4.3	Simulační modelování.....	49

4.3.1	Využití simulace.....	50
4.3.2	Výhody simulace.....	51
4.3.3	Nevýhody simulace	53
4.3.4	Tvorba simulačního modelu	54
4.3.5	Generování náhodných čísel v simulačních modelech.....	57
4.3.6	Druhy simulace.....	60
4.3.6.1	Discrete event simulace	60
4.3.6.2	System dynamics	62
4.3.6.3	Ostatní simulační metody	65
4.3.7	Význam simulace v modelování dodavatelských systémů.....	66
4.3.7.1	Význam simulačních metod v modelování dodavatelských systémů – srovnání s jinými modelovými přístupy	66
4.3.7.2	Srovnání významu jednotlivých simulačních metod	70
5	SIMULAČNÍ SOFTWARE	71
5.1	Software pro discrete event simulaci.....	74
5.1.1	Arena	75
5.1.2	Simul8	76
5.1.3	Witness	78
6	NÁVRH OBECNÉHO SIMULAČNÍHO MODELU MATERIÁLOVÝCH TOKŮ PRO OPTIMALIZACI STRUKTURY DODAVATELSKÝCH SYSTÉMŮ	84
6.1	Obecný simulační model materiálových toků vytvořený v prostředí Witness ...	85
6.2	Struktura obecného simulačního modelu materiálových toků	87
7	PŘÍKLADY APLIKACÍ OBECNÉHO SIMULAČNÍHO MODELU MATERIÁLOVÝCH TOKŮ	119
7.1	Redesign distribučního systému společnosti zabývající se výrobou a distribucí olejů a maziv.....	119
7.1.1	Současný stav struktury distribučního systému.....	119
7.1.2	Modelování distribučního systému.....	121
7.1.3	Posouzení efektivity navržené struktury distribučního systému – analýza nákladů na distribuci	130
7.1.4	Simulace současné struktury distribučního systému	132
7.1.5	Redesign distribučního systému	132
7.2	Návrh koncepce logistiky v centrálním skladu společnosti zabývající se nákupem a prodejem stavební chemie a obkladů.....	138
7.2.1	Současný stav logistiky v centrálním skladu společnosti	138

7.2.2	Návrh koncepce logistiky v centrálním skladu společnosti a její modelování pomocí obecného simulačního modelu materiálových toků	141
7.3	Optimalizace layoutu výrobní haly a návrh systému manipulace mezi výrobou a skladem společnosti zabývající se výrobou mazacích systémů	146
7.3.1	Současný layout výrobní haly a systém manipulace mezi výrobou a skladem; simulace současného stavu pomocí obecného simulačního modelu materiálových toků	146
7.3.2	Optimalizace layoutu výrobní haly a systému manipulace mezi výrobou a skladem s využitím obecného simulačního modelu materiálových toků	149
7.4	Reengineering procesů spojených s vychystáním objednávky zákazníka v centrálním skladu společnosti distribuující spotřební zboží	159
7.4.1	Současný stav logistiky v centrálním skladu společnosti	159
7.4.2	Změna systému vychystávání položek, reorganizace skladu	167
7.4.3	Stanovení efektivity navrženého systému vychystávání položek pomocí obecného simulačního modelu materiálových toků	168
8	ZÁVĚR	171
9	LITERATURA.....	174
	SEZNAM OBRÁZKŮ	195
	SEZNAM TABULEK	198
	SEZNAM POUŽITÝCH ZKRATEK.....	199
	REJSTŘÍK	200
	INFORMACE O AUTOROVÍ.....	203

1 ÚVOD

Postupující globalizace ekonomického prostředí, projevující se mimo jiné koncentrací kapitálu a kapacit, výrazným způsobem ovlivňuje strukturu materiálových toků. Přičteme-li k tomu prohlubující se diferenciaci trhů, vedoucí mnohdy až k výrobě výrobků koncipovaných podle přání individuálních zákazníků, je zřejmé, že je třeba razantním způsobem změnit nároky na všechny složky managementu a vytvořit vhodné prostředí, ve kterém lze na konkurenceschopné úrovni uspokojit požadavky konečných zákazníků. Vedení organizací stojí před problémy řídit výrobu a distribuci velkého množství výrobků ve stále větším počtu variant, v geograficky stále rozsáhlejších regionech dalece přesahujících rámec zemí. V konkurenci stále složitějších dodavatelských systémů postavených na spolupráci, vzájemné důvěře a vzájemném předávání informací mohou uspět jen ty subjekty, které nejenže využívají moderních manažerských metod řízení, ale jsou také schopny dynamicky měnit svou strukturu (Gros and Grosová, 2011).

Dynamická simulace představuje vynikající nástroj pro plánování, návrh a optimalizaci struktury dodavatelských systémů. Jedná se o metodu, která na jedné platformě umožňuje propojení procesů probíhajících při realizaci materiálových a informačních toků, jejich vizualizaci a optimalizaci s ohledem na nejrůznější strategie řízení a kritéria hodnocení výkonu. Simulace umožňuje práci s rozsáhlými soubory dat stochastické povahy, která jsou typická pro současné ekonomické prostředí, a to zejména v případě variabilní poptávky. Chybná rozhodnutí v oblasti řízení a plánování poptávky mají za následek velmi negativní dopady v rámci celého dodavatelského systému. V souvislosti s touto problematikou se nejčastěji hovoří o tzv. efektu biče (Forrester, 1958). Podstatou uvedeného jevu je fakt, že i malá změna poptávky na konečném trhu je postupným předáváním mezi jednotlivými články dodavatelského systému zkreslována (Lee, et al., 1997). K efektu biče (viz také bullwhip effect, Forresterův efekt, amplification effect) dochází tam, kde dodavatelský systém sdružuje řadu nezávislých podniků, jež společně sdílejí pouze minimum informací. Zpráva o malé změně spotřebitelské poptávky se proti směru řetězce zesiluje a výkyvy v poptávaných množstvích jsou obvykle tlumeny prostřednictvím rostoucích pojistných zásob. Důsledkem této situace jsou kromě nepřiměřené výše držené zásoby také značné výkyvy ve využití distribučních kanálů, zdrojů realizujících materiálové a informační toky či naprostá nefunkčnost automatizovaných systémů řízení zásob (Chen, et al., 1999). Jedná se o jevy, které s sebou přinášejí celou řadu skrytých nákladů a dalších rizik. Komplexnost dynamické simulace, možnost otestovat celou řadu různých uspořádání dodavatelského systému a jasně kvantifikovat jejich výkonnost předurčuje tuto metodu k využití v konceptech řízení materiálových toků založených na principu spolupráce v rozsáhlých a komplikovaných dodavatelských systémech. Aplikace těchto konceptů prokazatelně vede k redukci negativních vlivů řetězcových efektů, zvýšení flexibility dodavatelských systémů a úrovně služeb poskytovaných konečným zákazníkům při optimálním využití zdrojů, jež realizují materiálové a informační toky.

2 CÍLE A STRUKTURA MONOGRAFIE

Cílem této monografie je:

1. Popsat současný stav a kvantitativní přístupy používané při modelování dodavatelských systémů.
2. Na základě analýzy literatury pojednávající o modelování dodavatelských systémů vymezit význam dynamické simulace v porovnání s jinými modelovými přístupy a zhodnotit význam jednotlivých druhů simulace jejich vzájemným srovnáním.
3. V prostředí softwaru pro dynamickou simulaci Witness vytvořit obecný simulační model materiálových toků vhodný pro navrhování a optimalizaci struktury materiálových a informačních toků v dodavatelských systémech.
4. Na příkladech implementace obecného simulačního modelu materiálových toků zaměřených na návrh reálných dodavatelských systémů v různých průmyslových odvětvích ověřit funkčnost modelu a popsat jeho silné a slabé stránky.

Monografie je rozdělena na část teoretickou a praktickou.

V teoretické části je v kapitole 3 diskutováno současné pojetí logistiky, v němž organizace přechází od integrace vnitřních podnikových funkcí k vertikální a horizontální integraci v rámci rozsáhlých dodavatelských systémů. Je zde také popsán koncept supply chain managementu (SCM). V kapitole 4 jsou charakterizovány kvantitativní přístupy, které jsou v současnosti využívány při modelování dodavatelských systémů, a představeny základní typy simulací. Na základě analýzy literatury pojednávající o modelování dodavatelských systémů je stanoven význam dynamické simulace v porovnání s jinými modelovými přístupy a význam jednotlivých druhů simulace při porovnání vzájemném. Kapitola 5 shrnuje poznatky o simulačních softwarových produktech dostupných na současném trhu. Zvláštní pozornost je věnována produktům pracujícím na principu discrete event simulace jako Arena, Simul8 a Witness.

V praktické části je v kapitole 6 zformulován obecný model simulace materiálových toků vhodný pro navrhování a optimalizaci struktury materiálových a informačních toků v dodavatelských systémech. Tento model je vytvořen v prostředí softwaru pro dynamickou simulaci Witness s podporou MS Excel pro načítání vstupních dat a úpravu výstupů. Základním principem fungování obecného modelu je předpoklad, že každý materiálový tok lze rozložit do konečného počtu pohybů. Na jednoduchém příkladu simulace materiálového toku složeného ze dvou pohybů je diskutována struktura modelu rozdělená do 6 základních bloků. Každý blok je popsán z hlediska své funkce v modelu, přičemž pro fyzické a grafické elementy, jež jsou součástí bloků, jsou uvedeny Details, Rules a Actions zajišťující

správné provedení pohybů tvořících simulovaný materiálový tok. Details, Rules a Actions jsou popsány do detailu programového kódu vytvořeného pomocí programovacího jazyka Visual basic (VB). Kapitola 7 nabízí příklady implementace obecného simulačního modelu materiálových toků zaměřené na návrh reálných dodavatelských systémů v různých průmyslových odvětvích.

V závěru monografie jsou diskutovány výhody a nevýhody navrženého řešení.

3 SOUČASNÉ POJETÍ KONCEPCE LOGISTIKY

V osmdesátých letech 20. století dochází k zásadním změnám v koncepci logistiky. Tyto změny jsou vyvolány přechodem od trhu výrobců k trhu zákazníků. Stále větší individualizace potřeb konečných zákazníků, rostoucí požadavky na šíři sortimentu a komplikovanost výrobků, jejichž životní cyklus se neustále zkracuje, staví podniky před úkol plánovat a řídit velmi komplikované materiálové toky. Globalizace společnosti a trhů, koncentrace kapitálu a kapacit vede ke změně konkurenčního prostředí, kdy soutěž velkého množství individuálních podniků přechází v soutěž omezeného množství nadnárodních společností. V této souvislosti se ve vědecké a odborné literatuře i v manažerské praxi stále častěji prosazují nové pojmy. Tradiční základní pojmy jako logistický řetězec či logistický systém jsou nahrazovány termíny jako dodavatelský řetězec či dodavatelský systém.

3.1 Koncept supply chain

V zahraniční literatuře lze nalézt řadu definic pojmu supply chain (SC). Uvedme nyní některé z nich:

1. SC zahrnuje všechny kroky, které je třeba přímo nebo nepřímo uskutečnit pro splnění požadavků konečného zákazníka. SC nezahrnuje jen výrobce a dodavatele, ale i dopravce, sklady, prodejce a zákazníky. Prostřednictvím všech organizací, např. výrobců, obsahuje SC všechny funkce, které jsou nutné pro splnění požadavků zákazníků. Tyto funkce – a nejen ty – zahrnují vývoj nových výrobků, marketing, distribuci, financování a služby zákazníkům (Chopra and Meindl, 2007).
2. SC je charakteristický tokem materiálu od dodavatelů k zákazníkům a tokem informací od zákazníků k dodavatelům (Mentzer, et al., 2001).
3. SC je síť organizací, které jsou zapojeny po i proti směru materiálového toku do různých procesů a aktivit, které přinášejí hodnotu ve formě výrobků a služeb podle požadavků konečného zákazníka (Christopher, 2005).
4. V SC dochází k integraci skupiny podnikatelských subjektů, jako jsou dodavatelé, výrobci, distributoři atd. Jejich společným cílem je najít takové

řešení, které vede k efektivnímu uspokojování požadavků zákazníků, jako jsou široké portfolio nabízených produktů, vysoká kvalita a krátké dodací termíny (Stank, et al., 2001).

5. SC je síť partnerů, kteří kolektivně transformují komodity ve finální produkty s přidanou hodnotou pro konečného zákazníka a kteří na každém kroku realizují nezbytné zpětné toky. Každý partner přitom odpovídá za procesy přinášející hodnotu výrobkům (Harrison and Van Hoek, 2008).

Uvedené definice se shodují na základním cíli, orientaci na konečného zákazníka a důrazu na zvyšování hodnoty služeb a výrobků pro konečné zákazníky. Typické pro pojetí zahraničních autorů je v rámci SC sdružování jak aktivit, tak subjektů podílejících se na realizaci materiálových, informačních a peněžních toků.

Uveďme nyní pojetí SC některých českých a slovenských autorů:

1. Dodavatelský řetězec je integrovaným procesním logistickým řetězcem vedoucím od dodavatelů až ke konečnému zákazníkovi, resp. k recyklaci. Jde o posloupnost kroků přidávajících hodnotu, vedoucích k uspokojení konečného zákazníka, zprostředkovaných informačními technologiemi, dopravou, sklady atd. (Pernica, 2005).
2. Dodavatelský systém se skládá ze tří základních prvků:
 - dopravní, který je nositelem operace doprava, charakteristické změnou polohy prvku,
 - integrační, v němž dochází k operaci kumulace, u kumulovaného prvku dochází jen ke změně času a
 - transformační, v němž dochází k transformaci kvalitativních a kvantitativních parametrů transformovaného prvku (Malindžák, 2007).
3. Logistický řetězec je posloupnost činností, jejichž výkon je nezbytný pro splnění požadavků finálního zákazníka v požadovaném čase, množství, kvalitě a na požadované místo (Gros and Grosová, 2012).
4. Dodavatelský řetězec je horizontálně i vertikálně propojená množina logistických řetězců (Gros and Grosová, 2012).
5. Logistický systém je množina organizací a vazeb mezi nimi, jehož prvky se podílejí na plánování a výkonu posloupnosti činností v logistickém řetězci (Gros and Grosová, 2012).

6. Dodavatelský systém je horizontálně i vertikálně propojená množina logistických systémů (Gros and Grosová, 2012).

V pojetí českých a slovenských autorů je přechod od používání pojmu logistický k pojmu dodavatelský spojen s vývojem ekonomického prostředí, zejména s jeho postupnou globalizací, dále pak s individualizací služeb zákazníkům v důsledku rostoucí intenzity konkurence spojené s prohlubující se segmentací trhů. V Gros, et al. (2009) autoři formulovali rozdíly v pojetí logistického a dodavatelského řetězce:

- Ve srovnání s logistickým řetězcem se dodavatelský řetězec rozšiřuje po i proti směru materiálového toku.
- Koncepce dodavatelského řetězce v sobě zahrnuje aktivity spojené s realizací zpětných toků vrácených či použitých produktů, likvidací odpadů apod.
- Dodavatelské řetězce se transformují v dodavatelské sítě pomocí vzájemného vertikálního a horizontálního propojení.
- Jedním z projevů horizontální integrace je vzájemné propojení podnikových funkcí, jako je logistika, marketing, řízení výroby, řízení výzkumu a vývoje, řízení jakosti apod.
- Správná funkce dodavatelského řetězce není možná bez vzájemné důvěry, sdílení informací a spolupráce mezi partnery, kteří činnosti v řetězci realizují.

V rámci této monografie bude používán pojem dodavatelský systém ve smyslu pojetí konceptu SC zahraničními autory. To znamená, že nebudou důsledně rozlišovány činnosti, které jsou realizovány při uspokojování potřeb zákazníků, a subjekty, které se na této realizaci podílejí.

3.2 Supply chain management

Supply chain management (SCM) je definován jako plánování a řízení materiálových, informačních a peněžních toků v síti navzájem propojených organizací, které přidávají hodnotu výrobkům a službám s cílem uspokojení potřeb konečného zákazníka (Stock and Boyer, 2009). Z pohledu procesního zahrnuje SCM plánování, nákup, výrobu a distribuci, ale nezaměřuje se výhradně na jednu z těchto oblastí (Cooper, et al., 1997). Zahrneme-li do klasického pojetí SCM zaměřeného na ekonomickou výkonnost dodavatelských systémů sociální a environmentální aspekty (tzv. triple-bottom-line), hovoříme o sustainable supply chain managementu (SSCM) (Seuring and Müller, 2008). V tomto kontextu je SSCM zaměřen výhradně na SC bez zpětných toků, v případě jeho rozšíření o reverzní logistiku a recyklaci výrobků hovoříme o closed-loop SCM (Guide Jr and Van Wassenhove, 2009).

Mnozí autoři se shodují v tom, že základním principem SCM je spolupráce. Jedná se o aktivní přístup podniků k dosažení společného cíle (Mentzer, et al., 2000). Spolupráce mezi partnery v SC vede k:

- snížení nákladů a vyšší ziskovosti (Kalwani and Narayandas, 1995),
- redukci zásob, zvýšení úrovně služeb zákazníkům (Mentzer, et al., 2000),
- zintenzivnění přenosu informací mezi partnery, zvýšení přesnosti předpovědi poptávky (Wagner, et al., 2002),
- zvýšení prodejů, zkrácení termínu vyřízení objednávky (Simatupang and Sridharan, 2004),
- dosažení konkurenční výhody (Giaglis, et al., 2006).

Simatupang and Sridharan (2004) definují 4 základní stupně spolupráce:

- sdílení informací,
- společné plánování a rozhodování,
- sdílení nákladů a rizika a rozdělování efektů získaných spoluprací,
- společné investice.

Výše uvedené body jsou základem mnoha konceptů spolupráce v SC. Vendor managed inventory (VMI) představuje koncept spolupráce, kdy dodavatel disponuje oprávněním řídit zásoby odběratele. Jedná se o typ spolupráce, kdy sdílení informací o poptávce a aktuálním stavu zásob u odběratele probíhá pomocí informačních technologií typu electronic data interchange (EDI) nebo online protokolů na internetu (Yao, et al., 2007). Dodavatel poté sestavuje plán výroby a dodávek do skladů odběratele na základě principů hladinového řízení zásob. Úspěšnou aplikaci VMI představila například společnost Wal-Mart ve vztahu ke svému dodavateli Procter & Gamble (Çetinkaya and Lee, 2000).

Podstatou konceptu quick response (QR) je rychlý přenos informací o stavu zásob, vystavovaných objednávkách a uskutečněných prodejkách v celém systému od výrobců přes distributory až po maloobchodní prodejny. Předpokladem jeho implementace je elektronická identifikace pohybu zboží pomocí čárových kódů a efektivní přenos dat ve vysoké frekvenci ve velmi krátkých časových intervalech (denně) v prostředí EDI (Iyer and Bergen, 1997).

Reakce na problémy v distribuci potravin v USA a rostoucí problémy s náklady na činnosti, které nepřinášejí hodnotu pro zákazníka, stály u zrodu systému efektivní odezvy efficient customer response (ECR). Podstata ECR spočívá v navázání spolupráce partnerů v dodavatelském systému s cílem dosáhnout maximální efektivnosti při uspokojování potřeb konečných zákazníků. Metoda je zaměřena na tři hlavní oblasti činností (Kotzab, 1999):

- segmentaci výrobků a služeb,
- efektivní řízení promočních akcí,
- koordinaci aktivit spojených s uváděním nových výrobků na trh.

Segmentace výrobků a služeb probíhá na bázi různých požadavků zákazníků, přičemž dochází k vytvoření specializovaných distribučních systémů pro vytvořené segmenty tak, aby bylo v každém z nich dosaženo souladu mezi požadovanou úrovní služeb a náklady.

Efektivní řízení promočních akcí spočívá ve společném plánování, určení termínu zahájení, celkového trvání a lokalizace míst jejich konání. Plán umožňuje předběžnou přípravu akce, její logistické zabezpečení včetně odhadu velikosti prodeje, požadavků na dárky apod.

Konečně cílem koordinace aktivit spojených s uváděním nových výrobků na trh je omezit ztráty spojené s případným neúspěchem tohoto procesu. Metoda je postavena na spolupráci výrobců s dodavateli, distributory a zákazníky při vývoji nových výrobků a jejich uvádění na trh, přičemž spolupracující organizace se soustřeďují zejména na příčiny problémů se zaváděním nových výrobků na trh, jako je například společný odhad budoucí poptávky, vhodná cenová politika, zajištění potřebné výrobní a skladovací kapacity apod.

Změna modelu spolupráce obchodních partnerů a vytvoření významně přesnější informační základny, která povede hodnotový řetězec k vyšším prodejům a ziskům, je misí systému collaborative planning, forecasting and replenishment (CPFR). Základní principy tohoto systému lze shrnout do následujících bodů (Barratt and Oliveira, 2001):

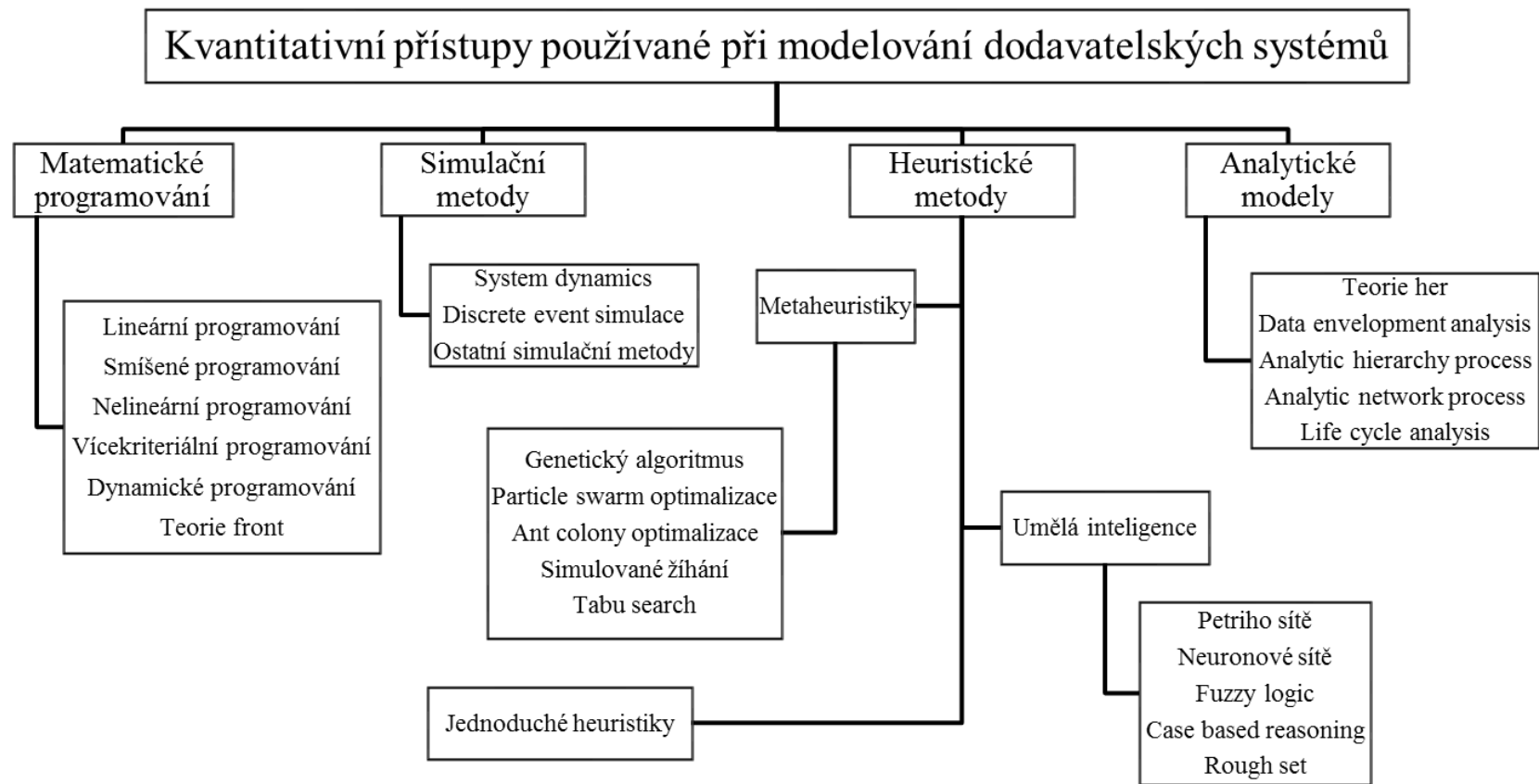
- společné plánování a řízení materiálového toku,
- společné cíle a metriky,
- dohoda o spolupráci,
- využití technologických standardů pro sdílení informací, dat, textů, zajištění bezpečnosti přenosu a zpracování dat,
- měření a vykazování společných efektů a dosažených výkonů dodavatelského systému,
- rozšiřování systému po i proti směru materiálového toku, horizontálně i vertikálně.

Mezi obecné zásady úspěšné implementace CPFR patří propojení procesů, nástrojů a lidí, společná orientace, angažovanost a soustředění na problém. Dále je nutné neorientovat se pouze na software a neopomíjet procesy, je třeba udržet a používat poznatky či je rozvíjet například pomocí školení top managementu i výkonných pracovníků. Velmi důležitá je vzájemná důvěra v informace, tj. že všichni udělají to, co říkají. Proto je třeba komunikovat s partnery vždy, když je to možné, neboť vědomosti znamenají pochopení procesů a pochopení vede k jejich správné a odpovědné realizaci.

4 SOUČASNÝ STAV MODELOVÁNÍ DODAVATELSKÝCH SYSTÉMŮ

4.1 Kvantitativní přístupy používané při modelování dodavatelských systémů

Při modelování materiálových a informačních toků v dodavatelských systémech se využívá řady kvantitativních přístupů. Tyto přístupy lze rozdělit do 5 hlavních skupin (viz Obr. 4.1).



Obr. 4.1 Kvantitativní přístupy používané při modelování dodavatelských systémů

4.1.1 Matematické programování

Matematické programování sdružuje modelové přístupy, které umožňují vybírat z množiny řešení nejlepší alternativu při respektování existujících omezení. V nejjednodušším případě se jedná o hledání minima či maxima funkce systematickým výběrem hodnot vstupních proměnných a výpočtu její hodnoty. Do této skupiny patří zejména lineární, nelineární, smíšené, vícekritériální a dynamické programování a také teorie front (Brandenburg, et al., 2014).

4.1.1.1 Lineární programování

Lineární programování patří mezi nejstarší a nepoužívanější modelové přístupy využívané v podnikové praxi. Aplikace lineárního programování v modelování a optimalizaci dodavatelských systémů jsou uvedeny např. v Hiremath, et al. (2013), Shabani, et al. (2014), Mousazadeh, et al. (2015), Kim and Kim (2016) či Jensen, et al. (2017). Lineární programování se opírá o formulaci úlohy, která se skládá z účelové funkce, soustavy omezení a podmínky nezápornosti řešení. Tuto úlohu lze matematicky zapsat například v následující podobě (Dyntar and Gros, 2015):

$$\max z = \sum_{j=1}^n c_j x_j \quad (4.1)$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i \rightarrow i = 1, 2, \dots, k \quad (4.2)$$

$$\sum_{j=1}^n a_{ij} x_j = b_i \rightarrow i = k + 1, k + 2, \dots, k + p \quad (4.3)$$

$$\sum_{j=1}^n a_{ij} x_j \geq b_i \rightarrow i = k + p + 1, k + p + 2, \dots, k + p + s \quad (4.4)$$

$$x_j \geq 0 \rightarrow j = 1, 2, \dots, n \quad (4.5)$$

kde z představuje označení účelové funkce, x_j jsou optimalizované proměnné, c_j jsou ocenění optimalizovaných proměnných v účelové funkci, a_{ij} představují technické parametry modelu a b_i pravé strany omezení. Mezi optimalizované proměnné patří například vyráběná množství produktů, dále množství výrobků, které jsou dopravovány v síti zákazníků, aj. Ocenění optimalizovaných proměnných v účelové funkci jsou například prodejní ceny výrobků, sazby účtované dopravci za dopravu, náklady spojené s výrobou produktů, pracnost aj. Technické parametry modelu představují například nejruznější normy spotřeby, ukazatele spojené s kvalitou výrobků aj. Konečně pravé strany omezení představují nejčastěji nějaké kapacitní omezení spojené s maximálním disponibilním množstvím výrobních vstupů, maximální délkou disponibilního časového fondu či omezení vyplývající z požadavků trhu, jako je maximální či minimální prodané množství hotových výrobků. Pro řešení úloh lineárního programování se využívá simplexová metoda. Východiskem této metody je převedení úlohy lineárního programování

na tzv. standardní tvar. Toho lze docílit úpravou soustavy omezení na omezení typu rovno přidáním doplňkových proměnných. Dále jsou do omezení typu rovno a větší nebo rovno přidány pomocné proměnné s cílem zajistit podmínku nezápornosti řešení. Doplňkové a pomocné proměnné pak tvoří výchozí bazické řešení úlohy v podobě pravých stran omezení, přičemž ostatní proměnné úlohy jsou rovny 0. Simplexová metoda následně upravuje výchozí bazické řešení přidáváním a odebíráním proměnných tak, aby došlo k růstu (při maximalizaci) či poklesu (při minimalizaci) účelové funkce. Výhodou simplexové metody je nalezení globálního optima úlohy.

4.1.1.2 Celočíselné a smíšené programování

Existuje-li v úloze požadavek na celočíselnost všech optimalizovaných proměnných, hovoříme o celočíselném programování. Mohou-li hodnoty optimalizovaných proměnných nabývat pouze 0 a 1, jedná se o binární programování. V případě, že požadavek na celočíselnost platí pouze pro některé proměnné, jedná se o smíšené programování. Problémem úloh celočíselného a smíšeného programování je nalezení globálního optima úlohy. V případě využití simplexové metody k řešení úlohy je totiž nutné využít nějakého doplňkového přístupu, který zajistí fixaci hodnot optimalizovaných proměnných získaných simplexovou metodou (obecně jde o hodnoty neceločíselné) k celému číslu. Nejčastěji využívaným přístupem je tzv. metoda větví a mezí [viz. např. Viergutz and Knust (2014), Kalaitzidou, et al. (2015) či Petridis, et al. (2017)].

4.1.1.3 Nelineární programování

Pokud se v účelové funkci či v soustavě omezení objevují nelineární členy, jedná se o nelineární programování. Časté jsou situace, kdy je nutné optimalizovat nějaký poměrový ukazatel, jako je např. nákladovost tržeb či jejich rentabilita. V případě nákladovosti tržeb by účelová funkce úlohy měla následující tvar:

$$\min z = \frac{\sum_{j=1}^n n_{vj}x_j + N_f}{\sum_{j=1}^n c_jx_j} \quad (4.6)$$

kde x_j jsou prodávaná množství $j = 1, 2, \dots, n$ produktů, c_j jsou jednotkové prodejní ceny produktů, n_{vj} jednotkové variabilní náklady na výrobu produktů a N_f fixní náklady. Řešení úloh nelineárního programování je v některých případech možné linearizací úlohy a následným využitím simplexové metody. V případě, že účelová funkce či soustava omezení obsahuje konvexní nebo konkávní funkce, je možné řešení pomocí Khun-Tuckerovy věty o sedlovém bodě (Ozdemir and Cho, 2016). Stejně jako v případě lineárního programování lze v literatuře najít celou řadu prací zabývajících se využitím nelineárního programování k modelování a optimalizaci dodavatelských systémů. Jedná se např. o práce Zhang and Wright (2014), Garcia-Caceres, et al. (2015), Yang, et al. (2016) či Azadeh, et al. (2017).

4.1.1.4 Vícekriteriální programování

Při formulaci modelů rozhodovacích situací vzniká často požadavek na hledání řešení vyhovujícího více stanoveným kritériím. Například výrobní program společnosti je účelné postavit tak, aby došlo nejen k maximalizaci tržeb, ale také k vytvoření odpovídajícího zisku, spotřebě určitého množství nákladů či minimalizaci určitých kvantifikovatelných dopadů výrobního procesu na životní prostředí. Všechny uvedené důvody vedly k rozvoji vícekriteriálního programování. V případě, že účelová funkce modelu vícekriteriálního programování obsahuje pouze lineární členy, hovoříme o lineárním vícekriteriálním programování a obecný matematický model je možné zapsat ve tvaru (Dyntar and Gros, 2015):

$$\max z_s = \sum_{j=1}^n c_{sj}x_j \rightarrow s = 1, 2, \dots, k \quad (4.7)$$

$$\sum_{j=1}^n a_{ij}x_j \leq b_i \rightarrow i = 1, 2, \dots, m \quad (4.8)$$

$$x_j \geq 0 \rightarrow j = 1, 2, \dots, n \quad (4.9)$$

kde k je počet zvolených kritérií účelových funkcí.

Většina metod výběru řešení využívá v nějaké míře preferenční informace poskytnuté rozhodovatelem. Jedná se o dodatečné informace o tom, jakým kritériím dává rozhodovatel přednost na kritériální množině tvořené množinou bodů přiřazených všem přípustným řešením v euklidovském k -rozměrném prostoru se souřadnicemi $z_s(x)$ pro $s = 1, 2, \dots, k$. Nalezenému řešení úlohy se říká kompromisní. Metoda globální účelové funkce je založena na vytvoření agregované preferenční účelové funkce $Z(x) = [z_1(x), z_2(x), \dots, z_k(x)]$ zavedené na kritériální množině (Hwang and Masud, 2012). Agregovaná preferenční účelová funkce má obvykle tvar:

$$Z = \sum_{s=1}^k z(x)_s v_s \quad (4.10)$$

kde v_s jsou váhy přiřazené jednotlivým účelovým funkcím, které splňují podmínku:

$$\sum_{s=1}^k v_s = 1 \rightarrow 0 \leq v_s \leq 1; s = 1, 2, \dots, k \quad (4.11)$$

Je zřejmé, že váhy vyjadřují relativní významnost optimalizovaných kritérií, které jim rozhodovatel přisuzuje.

Lexikografická metoda pracuje s pořadím významnosti jednotlivých optimalizovaných kritérií dle preferencí rozhodovatele. Princip lexikografické metody spočívá v řešení posloupnosti úloh s jedním optimalizovaným kritériem, přičemž v každém kroku je připojena podmínka, že hodnota kritéria optimalizovaného v předchozím kroku neklesne pod ideální hodnotu, respektive v každém dalším kroku je přípustná pouze předem stanovená odchylka od získané hodnoty kritéria v kroku předcházejícím (Harzheim, 2006).

Aplikace lineárního vícekritériálního programování v modelování a optimalizaci dodavatelských systémů jsou uvedeny např. v Almaraz, et al. (2013), Zhang and Reimann (2014), Azadeh, et al. (2015), Cambero and Sowlati (2016) či Jindal and Sangwan (2017).

4.1.1.5 Dynamické programování

Pro dynamické programování je charakteristické přijímání rozhodnutí podle vývoje reálného systému v čase (Bellman, 1954). Jde například o situace, kdy je třeba rozhodnout o výměně výrobního zařízení, rozdělit omezené zdroje mezi divize podniku apod. Typické pro řízení takových situací a procesů, které je tvoří, je rozhodování po etapách (Dyntar and Gros, 2015). Etapa představuje ucelenou část procesu a lze ji definovat jako časovou či věcnou. Časová etapa je například rok, týden či měsíc, věcnou etapou může být například divize podniku, zeměpisná lokalita či úsek produktovodu. V praktických úlohách obvykle platí, že počet etap je konečný. Každá etapa je ohraničena počátečním a konečným stavem, které lze popsat vektorem stavových proměnných. Stejně jako u počtu etap obvykle platí předpoklad, že počet stavů a proměnných, které je charakterizují, je konečný. Při přijímání rozhodnutí v jednotlivých etapách je důležitá volba strategie. Strategie převádí systém z počátečního stavu v etapě do stavu konečného, přičemž dochází ke generování přínosu, v případě přínosu v jedné etapě jde o přínos dílčí. Cílem dynamického programování je potom pomocí volby strategií převést systém z počátečního stavu první etapy do konečného stavu etapy poslední tak, aby bylo dosaženo maximálního (např. v případě zisku) či minimálního (např. v případě nákladů) celkového přínosu, který je součtem přínosů dílčích. Jedná se tedy o nalezení optimální posloupnosti strategií s cílem optimalizovat celkový přínos. Zajímavý z pohledu hledání optimální posloupnosti strategií je postup od konce. To znamená, že pro proces, který se skládá z $i = 1, 2, \dots, n$ etap, je optimalizována nejprve poslední tj. n -tá etapa. Je to z toho důvodu, že neproběhne-li optimálně poslední etapa, neproběhne optimálně ani celý proces. Problémem je ovšem fakt, že není znám počáteční stav etapy, který je závislý na tom, co se stalo v etapách předchozích. Proto je nutné najít pro každý možný počáteční stav $x_{n-1, j}$ (viz důležitost konečného počtu možných stavů) takovou strategii $w_{n, j}$, aby platilo:

$$z_n^*(x_{n-1, j}) = \max/\min[z_n(x_{n-1, j}; w_{n, j})] \quad (4.12)$$

kde z_n^* představuje optimální celkový přínos v poslední tj. n -té etapě a z_n dílčí přínos v poslední tj. n -té etapě. Je zřejmé, že dílčí přínos je závislý na počátečním stavu a volbě strategie. Po optimalizaci poslední etapy následuje optimalizace předposlední a poslední etapy dohromady. Důvodem je předpoklad, že neproběhnou-li optimálně poslední dvě etapy, neproběhne optimálně ani celý proces. Problém opět představuje fakt, že není znám počáteční stav předposlední etapy, který závisí na tom, co se stalo v etapách předchozích. Proto je nutné najít pro každý možný počáteční stav předposlední etapy $x_{n-2, j}$ takovou strategii $w_{n-1, j}$, aby platilo:

$$z_{n-1,n}^*(x_{n-2,j}) = \max/\min\{z_{n-1}(x_{n-2,j}; w_{n-1,j}) + z_n^*[x_{n-1,j}(x_{n-2,j}; w_{n-1,j})]\} \quad (4.13)$$

a obecně pro optimalizaci etap i až n :

$$z_{i,n}^*(x_{i-1,j}) = \max/\min\{z_i(x_{i-1,j}; w_{i,j}) + z_{i+1,n}^*[x_{i,j}(x_{i-1,j}; w_{i,j})]\} \quad (4.14)$$

Uvedený postup se opakuje až do první etapy, která má obvykle jeden počáteční stav. Dynamické programování je vlastně numerická metoda, která v každém kroku (optimalizaci etapy) redukuje možný počet variant řešení tím, že nějakému výchozímu stavu etapy a vybrané strategii přiřazuje již známou optimální posloupnost strategií etap následujících. Uvažujme například situaci, kdy je třeba navrhnout optimální zásobovací strategii podniku klíčovou surovinou na následujících 5 měsících, přičemž odhad poptávaných množství této suroviny je 10; 18; 35; 87 a 24 tun. Nechť objednávky této suroviny mohou být realizovány vždy pouze na začátku měsíce na libovolný počet měsíců dopředu a cena jedné objednávky je 1 200 Kč. Dále nechť platí, že skladování jedné tuny suroviny stojí 85 Kč za měsíc. Z logiky dynamického programování popsané výše vyplývá, že pokud bychom chtěli uspokojit poptávku v pátém měsíci (viz postup optimalizace procesu od poslední etapy), bylo by nutné na začátku pátého měsíce objednat množství 24 tun, což znamená vynaložit náklady na objednání ve výši 1 200 Kč. Strategie objednat v pátém období na páté období je optimální strategií poslední etapy s optimálním celkovým přínosem v podobě nákladů na skladování a objednávání ve výši 1 200 Kč. Pokud bychom chtěli najít optimální strategii zásobování v měsících 4 a 5, máme dvě možnosti. První možnost je objednat v měsíci 4 pouze na měsíc 4 (87 tun) tj. vynaložit náklady na objednání ve výši 1 200 Kč. Tím se přesuneme na začátek měsíce 5, kde v předchozím kroku bylo zjištěno, že optimální strategií je objednat na měsíc 5 (24 tun) tj. opět vynaložit náklady na objednání ve výši 1 200 Kč. Celkem jsou tedy na objednávání v měsících 4 a 5 vynaloženy náklady na objednání a skladování ve výši 2 400 Kč. Druhá možnost je objednat v měsíci 4 takové množství suroviny, které pokryje poptávku v měsících 4 i 5 (87 + 24 = celkem 111 tun). To znamená vynaložení nákladů na objednání ve výši 1 200 Kč a skladování množství 24 tun, 1 měsíc za $1 \cdot 24 \cdot 85 = 2\,040$ Kč. Celkem jsou tedy na objednávání a skladování v měsících 4 a 5 vynaloženy náklady na objednání a skladování ve výši 3 240 Kč. Je zřejmé, že optimální posloupnost strategií uspokojení poptávky v měsících 4 a 5 představuje vystavení 2 objednávek v obdobích 4 a 5. Pro naznačený postup je charakteristické opakování velkého množství relativně jednoduchých výpočetních kroků, které mohou být jednoduše modelovány pomocí nějakého programovacího jazyka. Algoritmus hledání optimální strategie zásobování vytvořený v prostředí MS Excel pomocí jazyka Visual basic for applications může mít například následující podobu:

Sub DynProgr()

Dim PocetHodnot

```

Dim ns
Dim nj
Dim radek
Dim sloupec
Dim radekpomoc
Dim sloupecpomoc
Dim PocetOpakovani
Dim obdobi
Dim x
Dim DrzNaklady(10000, 1)
Dim DrzCestu(10000, 1) As String
PocetHodnot = Sheets("List1").Cells(1, 5).Value
nj = Sheets("List1").Cells(2, 5).Value
ns = Sheets("List1").Cells(3, 5).Value
radek = 6
x = 6
sloupec = 5
PocetOpakovani = PocetHodnot - 1
obdobi = 1
For aa = 1 To PocetHodnot
    Sheets("List1").Cells(5, aa + 4) = aa
    Sheets("List1").Cells(aa + 5, 4) = aa
    Sheets("List1").Cells(aa + 5, aa + 4) = nj
Next
While PocetOpakovani > 0
    For bb = 1 To 1
        Sheets("List1").Cells(bb + radek, sloupec) = nj + bb * Sheets("List1").Cells(bb + obdobi + 1, 2) * ns
    Next
    For bb = 2 To PocetOpakovani
        Sheets("List1").Cells(bb + radek, sloupec) = Sheets("List1").Cells(bb + radek - 1, sloupec) + bb * Sheets("List1").Cells(bb + obdobi + 1, 2) * ns
    Next
    PocetOpakovani = PocetOpakovani - 1
    sloupec = sloupec + 1
    obdobi = obdobi + 1
    x = x + 1

```

```

    radek = x
Wend
sloupec = 4
radek = 5
sloupecpomoc = 5
radekpomoc = 6
obdobi = 1
Sheets("List1").Cells(1, 7) = "f(1)*"
    For cc = 2 To PocetHodnot
        Sheets("List1").Cells(1, cc + 6) = "f(1," + CStr(cc) + ")*"
    Next
    For ab = 1 To PocetHodnot
        DrzNaklady(ab, 1) = 100000
    Next
DrzNaklady(1, 1) = nj
DrzCestu(1, 1) = CStr(Sheets("List1").Cells(radek, sloupecpomoc)) + "," +
CStr(Sheets("List1").Cells(radekpomoc, sloupec))
obdobi = obdobi + 1
While obdobi <= PocetHodnot
    If DrzNaklady(obdobi, 1) > Sheets("List1").Cells(obdobi + 5, 5) Then
        DrzNaklady(obdobi, 1) = Sheets("List1").Cells(obdobi + 5, 5)
        DrzCestu(obdobi, 1) = CStr(Sheets("List1").Cells(5, 5)) + "," +
CStr(Sheets("List1").Cells(obdobi + 5, 4))
    End If
    For dd = 1 To obdobi - 1
        If DrzNaklady(obdobi, 1) > DrzNaklady(dd, 1) + Sheets("List1").Cells(obdobi + 5, 5
+ dd) Then
            DrzNaklady(obdobi, 1) = DrzNaklady(dd, 1) + Sheets("List1").Cells(obdobi + 5,
5 + dd)
            DrzCestu(obdobi, 1) = DrzCestu(dd, 1) + "+" + CStr(Sheets("List1").Cells(5, dd +
5)) + "," + CStr(Sheets("List1").Cells(obdobi + 5, 4))
        End If
    Next
    obdobi = obdobi + 1
Wend
For ee = 1 To PocetHodnot
    Sheets("List1").Cells(2, ee + 6) = DrzNaklady(ee, 1)

```

Sheets("List1").Cells(3, ee + 6) = DrzCestu(ee, 1)

Next

End Sub

Aplikací popsaného kódu v prostředí MS Excel lze snadno a velmi rychle nalézt optimální strategii zásobování pro uvedený příklad. Tato strategie spočívá v objednávání v každém z pěti měsíců s celkovými náklady na skladování a objednávání ve výši 6 000 Kč. Stejně jako v případě lineárního a nelineárního programování lze v literatuře najít celou řadu prací zabývajících se využitím dynamického programování k modelování a optimalizaci dodavatelských systémů. Jedná se např. o práce Yang, et al. (2007), Tsao, et al. (2013), Abdulwahab and Wahab (2014) či Wang and Nguyen (2017).

4.1.1.6 Teorie front

Cílem teorie front je modelování procesu uspokojování požadavků na obsluhu a nalezení optimální struktury systému, do kterého tyto požadavky vstupují (Dyntar and Gros, 2015). Typickým jevem při realizaci těchto procesů je vznik front požadavků čekajících na obsluhu, jakmile je intenzita obsluhy nižší než intenzita vstupu požadavků do systému. Příkladem jednoduchého obslužného systému může být například příjezd automobilů na čerpací stanici s cílem doplnit pohonné hmoty či zákazníci přicházející do provozovny rychlého občerstvení za účelem nákupu oběda. Modelování obslužných systémů pomocí teorie front má obvykle poskytnout odpovědi na následující otázky:

- Jaký počet požadavků na obsluhu lze očekávat v průběhu času?
- Jaká bude potřeba obslužných míst?
- Jaké bude využití obslužných míst?
- Jaká bude úroveň služeb zákazníkům a zajistí tato úroveň konkurenceschopnost provozu?
- Jaké budou investiční a provozní náklady systému?

Z předchozího popisu vyplývá, že základním prvkem teorie front je požadavek na obsluhu. Požadavky na obsluhu náhodně vstupují do obslužného systému v průběhu času a vytvářejí vstupní proud požadavků. Vstupní proud požadavků na obsluhu je omezený či neomezený. Omezenost vstupního proudu požadavků je spojena s maximálním počtem požadavků na obsluhu, který může do systému vstoupit. V praxi jde například o systémy údržby, ve kterých je požadavkem na obsluhu porucha stroje a maximálně se může porouchat právě tolik strojů, kolik jich je v podniku k dispozici. Požadavky na obsluhu se dále řadí před obslužnými místy do front a z front vstupují do procesu obsluhy, který realizují obslužná místa. Způsob, jakým požadavky na obsluhu vstupují z fronty na obslužná místa, se nazývá disciplína fronty (Doytchinov, et al., 2001). Mezi disciplíny fronty patří například:

- first in first out (FIFO),
- last in first out (LIFO),
- náhodný výběr požadavků na obsluhu z fronty,
- výběr požadavků na obsluhu z fronty na základě priorit.

Další vlastností fronty je omezenost či neomezenost. Neomezená fronta znamená, že počet prvků ve frontě není omezen. V případě fronty omezené je tento počet omezen například z důvodů prostorových. Počet obslužných míst je roven 1 či větší než 1 a místa jsou řazena vedle sebe či za sebou. Obsloužené požadavky poté opouští systém v podobě výstupního proudu. Stejně jako pro vstupy požadavků na obsluhu do systému i pro trvání obsluhy zpravidla platí, že jde o náhodnou veličinu. V literatuře lze nalézt velké množství prací, ve kterých je k popisu náhodných vstupů požadavků na obsluhu do systému a náhodného trvání obsluhy využito Poissonova, Erlangova či exponenciálního rozdělení [viz např. McManus, et al. (2004), de Bruin, et al. (2007), Aksin, et al. (2007), Shin and Moon (2014), Wiecek, et al. (2016) či Takagi (2017)]. V případě, že k popisu vstupu požadavků na obsluhu do systému je využito Poissonova rozdělení, je náhodnou veličinou počet vstupů požadavků n v časovém intervalu o konstantní délce (např. ΔT) a tuto veličinu lze popsat následující rovnicí pro hustotu pravděpodobnosti:

$$f_n(\Delta T) = \frac{1}{n!} [(\lambda \Delta T)^n e^{-\lambda \Delta T}] \quad (4.15)$$

kde λ je intenzita vstupu požadavků vyjadřující průměrný počet požadavků na obsluhu, který vstoupí do systému obsluhy za jednotku času. Intenzitu vstupu požadavků lze pro Poissonovo rozdělení získat z rovnice pro průměr \bar{n} v následující podobě:

$$\bar{n} = \lambda \Delta T \quad (4.16)$$

V případě, že je k popisu vstupu požadavků na obsluhu do systému využito exponenciálního rozdělení, je náhodnou veličinou čas mezi 2 následujícími vstupy požadavků na obsluhu do systému t a tuto veličinu lze popsat následující rovnicí pro hustotu pravděpodobnosti:

$$f(t) = \lambda e^{-\lambda t} \quad (4.17)$$

kdy intenzitu vstupu požadavků lze pro exponenciální rozdělení získat z rovnice pro průměr \bar{t} v následující podobě:

$$\bar{t} = \frac{1}{\lambda} \quad (4.18)$$

Analogicky lze pro popis náhodného trvání obsluhy t' použít exponenciálního rozdělení s hustotou pravděpodobnosti v následujícím tvaru:

$$f(t') = \mu e^{-\mu t'} \quad (4.19)$$

kde μ je intenzita obsluhy vyjadřující průměrný počet požadavků, který systém obslouží za jednotku času. Intenzitu obsluhy lze pro exponenciální rozdělení získat z rovnice pro průměr \bar{t}' v následující podobě:

$$\bar{t}' = \frac{1}{\mu} \quad (4.20)$$

Schopnost systému obsluhovat vstupující požadavky na obsluhu pak popisuje intenzita provozu ρ jako:

$$\rho = \frac{\lambda}{\mu} \quad (4.21)$$

Začne-li nějaký obslužný systém, ve kterém je vstup požadavků a jejich obsluha popsána exponenciálním rozdělením, pracovat v čase 0 a zvolíme-li dostatečně malý interval ΔT tak, aby v něm došlo maximálně k 1 vstupu požadavku na obsluhu a maximálně k 1 obsluze, pak pravděpodobnost, že v nějakém čase $T + \Delta T > 0$ bude v systému právě $n > 0$ požadavků na obsluhu [tj. $p_n(T)$], lze popsat soustavou diferenciálních rovnic v následující podobě (Dyntar and Gros, 2015):

$$\frac{dp_n(T)}{dT} = \lambda p_{n-1}(T) - (\lambda + \mu) p_n(T) + \mu p_{n+1}(T) \rightarrow n > 0 \quad (4.22)$$

Pravděpodobnost, že v nějakém čase $T + \Delta T > 0$ bude v systému právě $n = 0$ požadavků na obsluhu [tj. $p_0(T)$], je:

$$\frac{dp_0(T)}{dT} = -\lambda p_0(T) + \mu p_1(T) \quad (4.23)$$

Uvedené diferenciální rovnice popisují nestacionární stav obslužného systému, kdy pravděpodobnosti, že v systému bude v nějakém okamžiku určitý počet požadavků na obsluhu, kolísají, respektive jsou závislé na čase. To znamená, že pro řešení soustavy diferenciálních rovnic by bylo nutné znát výchozí stav charakterizovaný počtem požadavků na obsluhu v systému. Budeme-li ovšem uvažovat, že intenzita provozu je menší než 1 (tj. intenzita obsluhy je vyšší než intenzita vstupu požadavků do systému) a že obslužný systém bude v provozu dostatečně dlouho, dochází k přechodu systému do stacionárního stavu, kdy pravděpodobnosti $p_n(T)$ přestávají být závislé na čase (nabývají svého maxima) a diferenciální rovnice lze položit rovné 0. Stacionární stav obslužného systému je zajímavý, neboť charakterizuje jeho chování po počáteční stabilizaci. To znamená, že lze relativně

snadno spočítat charakteristiky systému, které poskytují odpovědi na otázky související s délkou fronty a času, který v této frontě požadavky na obsluhu stráví, s využitím obslužných míst či investiční a provozní náklady spojené s pořízením a provozem systému. Postup výpočtu pro systémy s paralelním uspořádáním obslužných míst a exponenciálním rozdělením popisujícím náhodné vstupy požadavků do systému a náhodné trvání obsluhy je následující:

1. Identifikace počtu obslužných míst, omezení vstupního proudu požadavků a omezení fronty.
2. Stanovení intenzity vstupu požadavků a intenzity obsluhy na základě sledování systému či pomocí odhadu.
3. Stanovení intenzity provozu.
4. Výpočet pravděpodobnosti $p(n)$ jako funkce $p(0)$.
5. Výpočet $p(0)$ na základě předpokladu, že součet všech pravděpodobností je roven 1.
6. Využití pravděpodobností $p(n)$ ke stanovení charakteristik obslužného systému.

Mezi požadované charakteristiky obslužného systému patří:

- průměrný počet požadavků na obsluhu v systému,
- průměrná délka fronty,
- průměrný čas, který stráví požadavek na obsluhu v systému,
- průměrný čas, který stráví požadavek na obsluhu ve frontě,
- průměrný počet nevyužitých míst,
- průměrné využití obslužného místa.

Aplikace teorie front v modelování a optimalizaci dodavatelských systémů jsou uvedeny např. v Moghaddam and Nof (2014), Gong, et al. (2015), Gottlich and Kuhn (2016) či Yousefi-Babadi, et al. (2017).

4.1.2 Heuristické metody

Velmi zajímavou skupinu představují heuristické metody. Jedná se o modely, které obsahují prvky umělé inteligence. Umělá inteligence sdružuje poznatky z oblastí jako statistika, matematická optimalizace, logika, ekonomie, psychologie, informatika a mnoho dalších (Weiss, 1999). Jedná se o optimalizační metody se schopností „učit se...“. Patří sem

například Petriho a neuronové sítě, fuzzy logic a metaheuristické algoritmy jako genetický algoritmus, ant colony optimalizace, particle swarm optimalizace, algoritmus simulovaného žíhání či tabu search.

4.1.2.1 Petriho síť

Petriho síť je matematická reprezentace diskretních distribuovaných systémů. Byla navržena a popsána C. A. Petrim v roce 1962 [viz Petri (1962)]. Petriho síť se nejčastěji využívá k modelování obslužných systémů. Tato síť graficky reprezentuje strukturu distribuovaného systému jako orientovaný bipartitní graf s ohodnocením. V literatuře jsou popsány následující typy Petriho sítí (David and Alla, 2010):

- Condition/Event Petriho síť,
- Place/Transition Petriho síť,
- Place/Transition Petriho síť s prioritami,
- časované Petriho síť.
- hierarchické Petriho síť.

Condition/Event Petriho síť jsou tvořeny událostmi a podmínkami, které musí být splněny, aby určitá událost mohla nastat. Vazby mezi událostmi a podmínkami jsou znázorněny pomocí orientovaných hran. V grafickém znázornění Condition/Event Petriho sítí jsou podmínky zobrazeny zpravidla jako kroužky a události jako obdélníky či úsečky. Orientované hrany pak směřují buď od podmínky k události, nebo od události k podmínce. Platí, že podmínka je vstupní podmínkou události, pokud je zobrazena orientovanou hranou směřující od podmínky k události. Naopak podmínka je výstupní podmínkou události, jestliže je zobrazena orientovanou hranou směřující od události k podmínce. Velmi důležitou součástí Condition/Event Petriho sítí jsou tečky nazývané také tokeny. Tokeny jsou zakreslovány do kroužků označujících podmínky a vyjadřují jejich pravdivost. To znamená, že pokud je token v podmínce zakreslen, podmínka je splněna, pokud zakreslen není, podmínka splněna není. Počáteční rozložení tokenů v síti se nazývá počáteční značení, značení sítě je pak reprezentováno aktuálním rozložením tokenů v nějakém okamžiku, přičemž ke změně značení dochází uskutečňováním událostí. Pro události platí, že událost je proveditelná, jsou-li splněny všechny její vstupní podmínky a zároveň nesplněny všechny výstupní podmínky. V případě, že je proveditelná událost také provedena, dojde ke změně značení sítě a všechny vstupní podmínky jsou nesplněny (tj. jsou odebrány příslušné tokeny) a všechny výstupní podmínky jsou splněny (tj. jsou přidány tokeny).

Úpravou a rozšířením konceptu Condition/Event Petriho sítí byly vytvořeny Place/Transition Petriho síť. V Place/Transition Petriho sítích jsou podmínky nahrazeny místy a události přechody. Orientované hrany pak směřují buď od místa k přechodu, nebo od přechodu k místu. Významnou vlastností místa je jeho kapacita, která udává maximální

počet tokenů, které se mohou v místě v jeden okamžik nacházet. Dále je každé orientované hraně přiřazena váha, která udává, kolik tokenů se při provedení přechodu po dané hraně přesouvá. Stav sítě (značení sítě) je dán počty tokenů v jednotlivých místech sítě v určitém okamžiku. Analogicky s proveditelností události u Condition/Event Petriho sítě je u Place/Transition Petriho sítě proveditelnost přechodu podmíněna tak, že pro každé místo vstupní množiny přechodu platí, že obsahuje alespoň tolik tokenů, kolik činí váha hrany vedoucí z místa do přechodu, a že pro každé místo výstupní množiny přechodu platí, že počet tokenů obsažených v místě zvětšený o váhu hrany směřující z přechodu do místa nepřevyšuje jeho kapacitu. Realizací přechodu pak dochází ke změně stavu sítě tak, že v každém místě vstupní množiny přechodu se počet tokenů sníží o váhu hrany směřující z místa do přechodu a v každém místě výstupní množiny přechodu se počet tokenů zvýší o váhu hrany směřující z přechodu do místa. Při realizaci přechodu také dochází ke vzniku a zániku tokenů, neboť součet ohodnocení hran spojujících vstupní místa přechodu s přechodem se při modelování obslužných systémů velmi často liší od součtu ohodnocení hran spojujících výstupní místa přechodu s přechodem. Z uvedeného popisu vyplývá, že Condition/Event Petriho síť je vlastně speciálním případem Place/Transition Petriho sítě, kdy kapacita všech míst a váha všech hran je rovna jedné. Formální popis Place/Transition Petriho sítě lze využít například k modelování obslužného systému tvořeného vstupním proudem požadavků na obsluhu a obslužným místem, před kterým se tvoří fronta. Pomocí míst by bylo možné popsat přicházející požadavky, frontu a obsluhu, přechody by popisovaly vstup požadavku do systému, přechod požadavku z fronty do obsluhy a odchod požadavku ze systému.

Place/Transition Petriho síť s prioritami je Place/Transition Petriho síť, ve které je každému přechodu přiřazeno celé nezáporné číslo udávající prioritu přechodu. Priority přechodů upravují pravidla pro jejich provádění. V Place/Transition Petriho sítích s prioritami je přechod povolen, je-li proveditelný v odpovídající Place/Transition Petriho síti bez priorit. To znamená, že přechod je proveditelný za podmínek, kdy je povolen a žádný jiný povolený přechod nemá vyšší prioritu. Po provedení přechodu se značení sítě změní stejným způsobem jako v odpovídající Place/Transition Petriho síti bez priorit.

Všechny dosud zmíněné typy Petriho sítí mají jednu nevýhodu. Touto nevýhodou je fakt, že nepracují s časem a všechny změny v síti jsou provedeny okamžitě. Hovoříme-li o Place/Transition Petriho sítích, kde provedení přechodu odpovídá změně stavu systému, je často žádoucí, aby tyto změny stavu trvaly určitou dobu. Z tohoto důvodu byly navrženy časované Petriho sítě. Časované Petriho sítě mohou být deterministické, stochastické či kombinované, přičemž prvek času je spojen s různými částmi sítě. U sítí typu Transition-timed se časování projevuje tím, že provedení přechodu trvá určitou dobu, po kterou token pobývá uvnitř přechodu. U sítí typu Place-timed se časování projevuje tím, že token pobývá stanovenou dobu ve vstupním místě přechodu, jež má být proveden. Pro síť typu Arc-timed je typické, že přesun tokenu po příslušné hraně trvá určitou dobu. Konečně u Token-timed sítí je sice provádění přechodů v síti okamžité, ale tokeny opouštějící příslušný přechod jsou opatřeny časovým razítkem, které udává, kdy může být daný token zase

použit. Hodnota časového razítka odpovídá aktuální hodnotě simulačního času zvětšenou o příslušnou hodnotu.

Hierarchické Petriho sítě umožňují členit vytvářenou síť na jednotlivé podsítě, které jsou navzájem propojeny. Hierarchickou Petriho sítí se rozumí částečně uspořádaná množina nehierarchických Petriho sítí (tzv. stránek), přičemž platí, že stránka B je pod stránkou A, jestliže síť na stránce B rozvíjí některý prvek ze stránky A. Za tímto účelem se využívají hierarchizační konstrukty jako substituce přechodů (tj. přechod v dané síti je nahrazen substituující sítí), substituce míst (tj. místo v dané síti je nahrazeno substituující sítí), volání přechodů, slučování přechodů a slučování míst.

Aplikace Petriho sítí v modelování a optimalizaci dodavatelských systémů jsou uvedeny např. v Liu and Xu (2008), Liu, et al. (2009), Hsieh and Lin (2014), Latorre-Biel, et al. (2015) či Azougagh, et al. (2016).

4.1.2.2 Neuronová síť

Koncept umělých neuronových sítí je inspirován biologickým nervovým systémem. Nervový systém lze rozdělit na centrální nervový systém, který představuje mozek a míchu, a periferní nervový systém představovaný zejména nervy. Hlavním posláním centrálního nervového systému je řídit organismus, zpracovávat signály, které se do něj šíří ze smyslových receptorů dostředivými drahami, a vydat na základě vyhodnocení tohoto signálu pokyn, který se šíří ve formě signálu odstředivými neuronovými drahami směrem k efektorům. V této aktivní dynamice centrálního nervového systému je možné pozorovat existenci vstupní informace, její zpracování procesní jednotkou a následné vygenerování výstupu, jenž je dále využit efektozem. Základní jednotkou umělé neuronové sítě je neuron. McCulloch and Pitts (1943) formulovali jednoduchý matematický model, který představoval teoretický model biologického neuronu. V tomto modelu měl každý neuron několik vstupů a jeden výstup. Vstupy byly rozděleny na excitační a inhibiční a model akceptoval pouze m binárních vstupů a jeden výstup. Princip modelu byl takový, že v případě, kdy na vstupu převládá excitační buzení nad inhibičním, je excitován i výstup neuronu. V opačném případě neuron excitován není. Dále v modelu platí, že vstupní synapse nedisponují vahami a slouží pouze pro přenos hodnoty. Hebb (1949) definoval pravidlo, které umožňovalo učit neuron změnou vah jeho vstupů. Vycházel z předpokladu, že pokud je neuron excitován korektně, dojde k posílení těch spojů, které vedly k excitaci. Naopak, pokud je neuron excitován nesprávně, je nutné tyto spoje oslabit. Rosenblatt (1958) popsal umělou neuronovou síť tzv. perceptron. Tato síť sloužila k rozpoznávání znaků promítaných na plátno. Widrow (1960) provedl zobecnění základních principů matematického modelu neuronu a navrhl adaptivní lineární element sloužící k efektivnějšímu učení neuronu. Na přelomu 80. a 90. let 20. století pak došlo k nejrozsáhlejšímu rozvoji konceptu umělých neuronových sítí [viz např. Hecht-Nielsen (1988), Becker and Le Cun (1988) či Ash (1989)].

Umělá neuronová síť je tvořena matematickými neurony, primitivními jednotkami, z nichž každá zpracovává vstupní signály s určitými váhami a generuje výstup. Neuronová síť představuje topologické uspořádání jednotlivých neuronů do struktury komunikující pomocí orientovaných ohodnocených spojů. Každý umělá neuronová síť je charakterizována typem neuronů, jejich topologickým uspořádáním a strategií adaptace při trénování, učení sítě. Nejčastějším typem neuronové sítě je dopředná síť, ve které se signál šíří sítí od vstupu jednosměrně směrem k výstupům sítě (Biganzoli, et al., 1998). Jednotlivé neurony dopředné sítě jsou uspořádány do vrstev, přičemž mezi neurony v rámci jedné vrstvy neexistuje propojení, naopak mezi neurony sousedních vrstev zpravidla existuje propojení úplné. Spoje mezi neurony, které představují dráhy pro šíření signálu, jsou orientovány a každý spoj je ohodnocen vahou, která modifikuje intenzitu procházejícího signálu. Neexistenci konkrétního spoje lze modelovat spojením s vahou rovnou nule. První vrstva dopředné sítě se nazývá vstupní vrstva, poslední vrstva se nazývá vrstva výstupní a ostatní vrstvy se nazývají vrstvy skryté. Dopředné neuronové sítě jsou obvykle tvořeny jednou či dvěma skrytými vrstvami (Šíma and Neruda, 1996). Dopředná neuronová síť představuje masivně paralelní distribuovaný dynamický systém, kde každá jednotka, neuron, pracuje lokálně, samostatně. Taková neuronová síť se vyznačuje značnou robustností a je odolná vůči poškození. Obvykle dokáže poskytovat relevantní výstupy i při poškození některých elementů. U neuronové sítě dané topologie je možné rozeznat dvě fáze aktivity, které se nazývají aktivní a adaptační dynamika. Aktivní dynamika je fází, během které síť na základě předložených vstupů produkuje výstupy, přičemž topologie a nastavení parametrů sítě zůstává zpravidla konstantní. Této fázi předchází fáze učení, kdy síť pomocí adaptačního algoritmu modifikuje své parametry a v některých případech i svou topologii.

Adaptační algoritmy lze rozdělit do dvou základních oblastí:

- učení bez učitele,
- učení s učitelem.

Pro učení bez učitele je typické, že adaptační algoritmus nemá k dispozici žádné kritérium správnosti hledané transformace vstupních dat. Pracuje na principu shlukování, ve vstupním prostoru dat hledá navzájem podobné elementy. Na základě parametrů předkládaných vzorků vstupních dat provádí jejich třídění do skupin, bez možnosti posouzení správnosti. Učení bez učitele je používáno například u neuronových sítí označovaných jako samostatně se organizující mapy (Kohonen, 1982).

V případě učení s učitelem disponuje adaptační algoritmus konečnou množinou dvojic vstupů a odpovídajících korektních výstupů řešené úlohy. To znamená, že algoritmus má k dispozici příklady správného chování a správné transformace z vektorů vstupních na vektory výstupní. Příklady správného chování mohou být získány například pozorováním a zaznamenáním vstupních a výstupních hodnot systému, který je předmětem modelování. Množina všech dostupných hodnot tedy představuje známou část chování systému a je pak využita adaptačním algoritmem k učení sítě a ověření její funkce. Množinu všech

dostupných dat je často účelné rozdělit na dvě části, a to na učící a testovací množinu. Samotné učení sítě probíhá iterativně: algoritmus jednotlivé prvky učící množiny předkládá postupně neuronu, zjišťuje jeho odezvu na předložený vstup a na základě odchylky jeho výstupu od výstupu požadovaného provádí korekci vah neuronu. Interval, ve kterém dojde k předložení všech vzorů učící množiny alespoň jednou, se jmenuje epocha učení. K dokončení učení sítě může být dle komplexnosti problému zapotřebí desítky až tisíce epoch. Okamžik zastavení učení lze definovat několika způsoby, a to například dosažením požadované malé chyby transformace nad učící množinou nebo dosažením maximálního počtu epoch. Za účelem vyhodnocení okamžiku, kdy je vhodné učení ukončit, je z učící množiny vyčleněna množina validační. V průběhu adaptace je nad touto množinou periodicky testována výkonnost nalezené transformace a při zastavení poklesu chyby nad validační množinou je proces adaptace ukončen. Výkonnost naučené neuronové sítě je pak po dokončení učení ověřena pomocí testovací množiny.

Kroky iterativního učení dopředné neuronové sítě s učitelem jsou:

1. Proveď zpracování vstupních dat.
2. Definuj učící, testovací a validační množiny.
3. Definuj strukturu neuronové sítě.
4. Proveď inicializaci vah neuronů pomocí generátoru náhodných čísel.
5. Nastav počet epoch učení na počáteční hodnotu $n = 0$.
6. Zahaj proces učení, nastav $n = n + 1$.
7. Pokud je počet provedených epoch učení $n \geq \max n$, ukonči učení a jdi na krok 14.
8. Realizuj n -tou epochu učení.
9. Vyber deterministicky nebo náhodně jeden vstupní vektor sítě z učící množiny.
10. Získej odezvu sítě, vyhodnoť chyby klasifikace na základě porovnání skutečného a předpokládaného výstupu.
11. Koriguj váhy neuronů sítě na základě získané chyby.
12. Pokud není epocha učení dokončena, jdi na krok 9.

13. Na konci epochy vyhodnot' chybu klasifikace přes celou učící množinu. Je-li chyba < požadovaná hodnota, ukonči učení a jdi na krok 14, v opačném případě jdi na krok 4.

14. Stanov úspěšnost činnosti sítě na testovací množině, pokud je nedostatečná, jdi na krok 4 nebo krok 3, případně zastav práci algoritmu.

V literatuře jsou popsány tyto způsoby učení neuronů a neuronové sítě:

- Hebbovo učení,
- Widrowovo učení.

Učení neuronové sítě podle Hebba je založeno na předpokladu, že pokud jsou dva sousední neurony aktivní současně, vazba mezi nimi se posiluje, zatímco při nesouhlasné aktivaci oslabuje. To znamená, že pokud je neuron excitován korektně, pak se posilují spoje, které tuto excitaci vyvolaly. Pokud je neuron excitován nekorektně, dochází k oslabení spojů, které tuto excitaci vyvolaly. A konečně pokud není neuron excitován, nic se neděje, tj. nedochází k úpravě vah.

Učení neuronové sítě podle Widrowa je založeno na geometrické interpretaci učení neuronu s binárním výstupem. Podstatou této interpretace je předpoklad, že neuron rozděluje v souladu s kapitolou výstupní prostor na dva poloprostory, a provádí tak klasifikaci vstupních vektorů do jednoho z nich. V případě, že je klasifikace daného vzoru ze vstupní množiny správná, není zapotřebí žádné úpravy vah. V případě, že je klasifikace chybná, leží klasifikovaný bod ve špatné části poloprostoru rozděleného dělicí nadrovinou.

Aplikace neuronových sítí v modelování a optimalizaci dodavatelských systémů jsou uvedeny např. v Huang, et al. (2008), Dong, et al. (2012), Arasteh, et al. (2014), Liu, et al. (2016) či Aalaei and Davoudpour (2017).

4.1.2.3 Fuzzy logic

Fuzzy logika je charakteristická tím, že umožňuje matematicky vyjádřit pojmy jako trochu, dost nebo hodně. Jinými slovy, fuzzy logika principiálně povoluje vyjádřit částečnou příslušnost k množině. Klasická teorie množin připouští pro příslušnost prvku do množiny pouze dvě možnosti – prvek buď do množiny patří, nebo nepatří. Každému prvku množiny je přiřazena hodnota tzv. charakteristické funkce množiny, která pro daný prvek množiny nabývá pouze dvou hodnot – hodnoty 1, jestliže prvek do množiny patří, a hodnoty 0, jestliže do množiny nepatří. Naproti tomu fuzzy množiny pracují s předpokladem, že nějaký prvek může do množiny patřit jen částečně (Zadeh, 1996). To znamená, že v teorii fuzzy množin může charakteristická funkce množiny (tj. funkce příslušnosti) nabývat všech hodnot od 0 do 1. Aplikovat principy fuzzy logiky může být výhodné při řízení složitých dynamických systémů. Pro aproximaci takových systémů a jejich vlastností tak, aby bylo možné tyto systémy efektivně řídit, je třeba provést následující kroky (Modrlák, 2004):

1. Změřit vstupní veličiny.
2. Zobrazit změřené veličiny ve vhodném měřítku na aplikovaná univerza.
3. Převést vstupní ostrá data na fuzzy data.
4. Nalézt výstupní fuzzy množinu.
5. Přiřadit – nalézt k výstupní množině vhodnou ostrou hodnotu akční veličiny.

Z výše uvedeného popisu vyplývá, že do fuzzy systému vstupují ostré hodnoty proměnných, tyto hodnoty jsou fuzzifikovány (tj. jejich průsečík s funkcí příslušnosti určuje pravdivostní hodnotu, s jakou daná ostrá hodnota patří do fuzzy množiny) a po nalezení výstupní fuzzy množiny a její defuzzifikaci je výstupem ostrá hodnota akční veličiny.

Pro fuzzifikaci jsou nejčastěji používány standardní funkce příslušnosti, mezi které patří trojúhelníková či lichoběžníková funkce. Typické pro fuzzy aproximaci je to, že aproximovaná a často nelineární funkce je popsána slovně pomocí fuzzy pravidel (Jura, 2005). Fuzzy pravidla vytváří expert, který dokáže daný problém vyjádřit slovním popisem (například v podobě výroků typu malý, dostatečně velký, střední apod.), na jehož základě je možné popsat problém pomocí fuzzy množin. Nevýhodou tohoto přístupu je skutečnost, že míra shody mezi chováním fuzzy systému a činností experta je závislá na správném a pokud možno úplném vytěžení znalostí. Další vlastností fuzzy aproximace je lokální citlivost. To znamená, že hodnoty aproximační funkce lze změnit jen v nějakém okolí pracovního bodu změnou konsekventu pravidla nebo změnou polohy či tvaru některé funkce příslušnosti.

Metody defuzzifikace lze rozdělit na empirické a heuristické. Tyto metody umožňují stanovení hodnoty akční veličiny v podobě nejlepšího kompromisu (metoda těžiště) nebo v podobě přijatelného řešení (metoda nejvýznamnějšího maxima).

Aplikace fuzzy logic v modelování a optimalizaci dodavatelských systémů jsou uvedeny např. v Lee and Lee (2012), Cinar, et al. (2014), Lee, et al. (2015), Ma, et al. (2016) či Soleimani, et al. (2017).

4.1.2.4 Case based reasoning

Základní východiska case based reasoning obsahují práce Schank (1983), Kolodner (1983) a Lebowitz (1983). Princip case based reasoning je založen na předpokladu, že řešení nových případů lze hledat pomocí řešení podobných případů v minulosti. Pro účely počítačového zpracování metody byl navržen formální koncept, který se skládá ze čtyř kroků (Aamodt and Plaza, 1994):

1. Vyhledej informace o minulých případech.

Na základě cíle, kterého má být dosaženo řešením problému, vyhledej relevantní případy, které s tímto souvisí. Případ se skládá z popisu problému, řešení a anotace, která charakterizuje, jak byl postup řešení odvozen.

2. Znovu použij řešení minulých případů.

Proveď mapování řešení případu z minulosti s ohledem na cílový problém. Přizpůsob řešení případu z minulosti tak, aby odpovídalo nové situaci.

3. Reviduj řešení minulých případů.

Po dokončení mapování řešení případu z minulosti a jeho úpravě na cílový stav otestuj nové řešení přímo na reálném problému nebo proveď simulaci. V případě vzniku nežádoucích účinků proveď další revizi.

4. Ulož úspěšné řešení nového případu.

Po úspěšném provedení revizí řešení minulých případů a dosažení uspokojivého cílového stavu ulož toto řešení nového případu do paměti ve struktuře: popis problému, řešení a anotace, která charakterizuje, jak byl postup řešení odvozen.

Aplikace konceptu case based reasoning v modelování a optimalizaci dodavatelských systémů jsou uvedeny např. v Tobail, et al. (2012), Anzaldi, et al. (2015) či Choy, et al. (2016).

4.1.2.5 Rough set

Rough set pracuje s klasickou teorií množin způsobem, kdy každá klasická množina je aproximována dvojicí množin, jejichž prvky do jisté míry zobecňují vlastnosti této množiny. Základy teorie množin rough set položil ve svých pracích Zdzisław Pawlak [viz Pawlak (1982), Pawlak (1985) a Pawlak, et al. (1988)]. Pomocí této teorie lze modelovat komplikované stochastické dynamické systémy. Základní principy rough set vycházejí z předpokladu, že systém je tvořen neprázdnou konečnou množinou objektů (tj. základní množinou), které lze popsat neprázdnou konečnou množinou atributů nabývajících určitých hodnot. Pokud bychom ze základní množiny objektů chtěli vybrat nějakou podmnožinu (tj. cílovou množinu) a tu charakterizovat pomocí nějaké vybrané podmnožiny atributů (tj. určit míru ekvivalence cílové množiny), využili bychom dle teorie rough set k tomuto účelu tzv. *P*-lower aproximaci a *P*-upper aproximaci. *P*-lower aproximace představuje množinu objektů, které lze jednoznačně přiřadit do cílové množiny. Tato množina se nazývá pozitivní oblast. *P*-lower aproximace je tedy konzervativní aproximace cílové množiny a skládá se z objektů, které v základní množině nemají žádné nevymezitelné klony vyloučené cílovou množinou a do cílové množiny patří s pravděpodobností rovné 1. *P*-upper

aproximace představuje množinu objektů, které lze možná přiřadit do cílové množiny. Z definice *P*-upper aproximace vyplývá, že rozdíl základní množiny objektů a množiny objektů vymezené touto aproximací dává množinu objektů, které lze určitě vyloučit jako členy cílové množiny. Hovoříme o negativní oblasti. *P*-upper aproximace je vlastně liberální aproximací cílové množiny, která zahrnuje všechny objekty, které mohou být členy cílové množiny s určitou pravděpodobností. Rozdíl *P*-upper a *P*-lower aproximace se nazývá hraniční oblast a jde o množinu objektů, které nemohou být ani přiřazeny do cílové množiny, ani z ní nemohou být vyloučeny. *P*-upper a *P*-lower aproximace tvoří rough set, jejíž přesnost je dána podílem počtu objektů tvořících *P*-lower aproximaci a *P*-upper aproximaci. Hodnoty této přesnosti se pohybují v intervalu $\langle 0,1 \rangle$ a vyjadřují, do jaké míry rough set reprezentuje cílovou množinu. Výhodou Pawlakova konceptu například oproti modelovým přístupům založeným na počtu pravděpodobnosti či fuzzy logice je, že není závislý na dodatečných informacích získaných od expertů, funkčních předpisech či expertních parametrech, které do modelování systémů vnášejí určitou míru subjektivity (Dütsch and Gediga, 1995). Z tohoto důvodu je teorie rough set užitečná k indukci pravidel z neúplných datových sad, hledání závislostí v datových sadách či pro matematické řešení nejasných konceptů.

Aplikace konceptu rough set v modelování a optimalizaci dodavatelských systémů jsou uvedeny např. v Che and Chiang (2012) či Guchhait, et al. (2015).

4.1.2.6 Genetický algoritmus

Genetické algoritmy pracují na principu optimalizace hodnot nezávislých proměnných pomocí adaptačního algoritmu inspirovaného evolucí v živé přírodě. Základ algoritmů této skupiny tvoří jedinci (Li, et al., 2017). Každý jedinec disponuje genotypem, který je nejčastěji reprezentován sekvencí informací (číselné či textové podoby) nebo programem, které představují prvky prostoru řešení (Pradenas, et al., 2013). Interpretací genotypu je tzv. fenotyp, což je konkrétní projev určitého genotypu (Li, et al., 2015). V přírodě jde například o modrou barvu očí člověka. Kvalita jedince je daná hodnotou tzv. fitness funkce, což je hodnota, která popisuje míru chyby danou vzdáleností kvality jedince od kvality optimální (Zhan, et al., 2009). Množina jedinců tvoří populaci, přičemž aktuální populace jedinců se nazývá generace. Adaptivní schopnost genetického algoritmu je založena na myšlence, že přirozeným výběrem kvalitních jedinců a jejich křížením dojde k postupnému řešení problému, tj. k nalezení lepších hodnot nezávisle proměnných. Proces postupné adaptace jedinců pomocí fitness funkce se v genetickém algoritmu nazývá evoluce. Tato evoluce je inspirována Darwinovou teorií a vychází z myšlenky, že lépe adaptovaní jedinci mají v určitém prostředí vyšší šanci přežít a reprodukovat se. Genetická výbava jedinců se přitom může odlišovat a k jejím změnám dochází náhodně a pouze v malé míře.

Činnost genetického algoritmu lze jednoduše popsat následující sekvencí kroků (Mitchell, 1998):

1. Inicializace

V tomto kroku dochází ke stanovení velikosti populace a generování jedinců tvořících výchozí generaci. Pro každého jedince je stanovena hodnota fitness funkce.

2. Selektce

Selektce slouží k výběru jedinců z populace, kteří se mohou křížit. Výběr těchto jedinců probíhá pomocí některé z výběrových metod, kterými jsou například:

- náhodný výběr,
- ořezávání,
- vážená ruleta.

Při náhodném výběru dochází k výběru jedinců z populace bez ohledu na jejich kvalitu danou hodnotou fitness funkce. K náhodnému výběru se nejčastěji využívá generování náhodných čísel od 0 do 1, tj. hodnot rovnoměrného rozdělení.

Pro metodu ořezávání je typické, že nejprve jsou všichni jedinci seřazeni dle hodnoty fitness funkce. Skupina seřazených jedinců je poté rozdělena na dvě části podle libovolně zvoleného parametru. Jedinci s nízkými hodnotami fitness funkce zařazení do druhé skupiny nemají šanci stát se předmětem selektce, jedinci z první skupiny jsou ke křížení vybíráni například aplikací náhodného výběru či principů vážené rulety.

Při použití vážené rulety k selekci jedinců je každému jedinci přiřazen úsek pomyslné rulety, který udává hodnota fitness funkce. S takto vytvořenou ruletou je poté provedeno losování jedinců ke křížení pomocí generování náhodných čísel rovnoměrného rozdělení. Čím kvalitnější je jedinec, tj. čím vyšší je hodnota jeho fitness funkce, tím vyšší je pravděpodobnost, že jedinec bude vybrán ke křížení.

3. Křížení

Při křížení dochází k výměně části genotypu dvou jedinců vybraných z populace při selekci. Při jednobodovém křížení si jedinci vyměňují pouze jednu informaci tvořící genotyp, při vícebodovém křížení množství libovolné. S určitou malou pravděpodobností se také může stát, že jedinci si při křížení

nevymění informaci žádnou a přežijí do další generace beze změny (Schaffer, et al., 1989).

4. Mutace

Mutace probíhá v rámci genotypu jedince. Je charakteristická tím, že s určitou malou pravděpodobností dojde k výměně informací uvnitř genotypu. V literatuře se obvykle udává výše pravděpodobnosti, se kterou dojde k mutaci, maximálně 5 % [viz Hynek (2008)]. Význam mutace spočívá v ochraně před příliš rychlou unifikací vlastností populace, která má za následek ztrátu potenciálně kvalitního genetického materiálu a předčasnou konvergenci kvality populace do nevyhovujícího stavu.

5. Vyhodnocení kvality generace po provedení selekce, křížení a mutace jedinců

Vyhodnocení kvality generace po provedení selekce, křížení a mutace jedinců probíhá opět na základě stanovení fitness funkce. Pokud při vyhodnocení dojde ke vzniku jedince, který splňuje požadavek na kvalitu genotypu, dochází k ukončení běhu algoritmu. Běh algoritmu může být také ukončen po dosažení předem definovaného počtu opakování kroků 2–5. V opačném případě algoritmus pokračuje krokem 6.

6. Náhrada stávající generace nově vygenerovanou

7. Návrat algoritmu do kroku 2, tj. selekce

Aplikace genetických algoritmů v modelování a optimalizaci dodavatelských systémů jsou uvedeny např. v Alfonso-Lizarazo, et al. (2013), Nasiri, et al. (2014), Sadeghi, et al. (2015), Yu and Solvang (2016) či Govindan, et al. (2017).

4.1.2.7 Ant colony optimalizace

Optimalizace ant colony je metaheuristika, která využívá poznatků o chování společenstev různých druhů mravenců k nalezení řešení aplikačních problémů patřících do třídy NP-úplné (Dorigo and Stutzle, 2003). Jedná se o obtížně řešitelné problémy, jejichž prostor řešení je natolik rozsáhlý, že lze pouze obtížně nalézt globální optimum v reálném čase. Z oblasti navrhování a optimalizace dodavatelských systémů patří mezi typické NP-úplné úlohy např. problém obchodního cestujícího. Optimalizace ant colony pracuje s myšlenkou, že hledání potravy mnoha druhů mravenců je založeno na nepřímé komunikaci zprostředkované prostředím. Na cestě mezi mraveništěm a zdrojem potravy totiž mravenci vytvářejí chemickou stopu v podobě feromonů. Tu ostatní mravenci cítí a s vyšší

pravděpodobnostní volí spíše cestu se silnou koncentrací feromonů, čímž dojde k jejímu opětovnému posílení. Pokud ovšem feromonová stopa není udržována, vyprchá a zmizí.

Aplikace ant colony optimalizace v modelování a optimalizaci dodavatelských systémů jsou uvedeny např. v Xiao, et al. (2012), Validi, et al. (2014), Lagos, et al. (2015), Zohal and Soleimani (2016) či Zhang, et al. (2017).

4.1.2.8 Particle swarm optimalizace

Particle swarm optimalizace je metaheuristika inspirovaná chováním hejna ptáků či ryb při hledání potravy. Poprvé byla popsána Kennedym a Eberhartem v roce 1995 [viz Kennedy and Eberhart (1995)]. Tato metaheuristika vychází z existence určité populace částic, které tvoří roj směřující ke globálnímu optimu. Roj se pohybuje určitým směrem, jenž je částečně narušován nepředvídatelným odbočováním. V principu je směr pohybu roje v každém časovém okamžiku nejvíce ovlivňován optimem momentálně nejlepší z částic. Původně byla tato technika vyvinuta s cílem zkoumat společenské biologické systémy, chování jedinců tvořících kolektiv a jejich vzájemné interakce s okolím. V současnosti se tato metaheuristika, stejně jako ant colony optimalizace, využívá k řešení aplikačních problémů patřících do třídy NP-úplné.

Aplikace particle swarm optimalizace v modelování a optimalizaci dodavatelských systémů jsou uvedeny např. v Kadadevaramath, et al. (2012), Sadeghi, et al. (2014), Gueller, et al. (2015), Kumar, et al. (2016) či Maiyar and Thakkar (2017).

4.1.2.9 Simulované žíhání

Algoritmus simulovaného žíhání vychází z analogie s fyzikálním procesem ochlazování taveniny. Žíhání je termický proces, při kterém se pevná látka nejprve roztaví a potom se ochlazováním převádí do stabilního krystalického stavu, který je charakterizován minimální energií. Není-li počáteční teplota dostatečně vysoká nebo není-li ochlazování správně řízeno, nedojde v systému po každém snížení teploty k dosažení rovnovážného stavu, ale ke vzniku metastabilních krystalických struktur. Jak vyplývá z Boltzmannova distribučního zákona, který charakterizuje tepelnou rovnováhu, může i při nízké teplotě s určitou pravděpodobností dojít k přechodu do energeticky náročnějšího stavu (Moore, 1981). Právě díky těmto přechodům ovšem systém neuvázne v lokálním metastabilním stavu. Metoda simulovaného žíhání vychází z algoritmu lokální optimalizace známého také pod označením down-hill či descent algoritmus (Lenstra, 2003). Princip tohoto algoritmu je pomocí pseudokódu popsán níže:

Lokální prohledávání

Start

Generuj počáteční stav y^i

Opakuj

Generuj řešení y^j v okolí y^i

$$F_{ij} = F(y^j) - F(y^i)$$

Když $F_{ij} < 0$, tak $y^j = y^i$

dokud není $F_{ij} \geq 0$ pro všechna y^j z okolí bodu y^i

Konec

Algoritmus lokální optimalizace vychází z nějakého počátečního stavu y^i a pomocí vhodné procedury generuje řešení y^j , které s počátečním stavem sousedí. Následuje vyhodnocení změny účelové funkce F_{ij} odpovídající přechodu systému ze stavu i do stavu j . Dojde-li k poklesu účelové funkce (v případě hledání minima), přepíše se výchozí stav y^i řešením y^j . Práce algoritmu skončí v případě, že nelze nalézt žádné zlepšení v okolí aktuálně dosaženého nejlepšího řešení. Nedostatek takového přístupu je zřejmý – systém uvázne v prvním lokálním optimu. Cest, jak zlepšit činnost této procedury, je několik. První je rozšíření prohledávaného okolí aktuálního řešení, což ovšem vede k nárůstu výpočetního času. Druhou možností je spustit prohledávání z více počátečních stavů a z dosažených lokálních optim vybrat to nejlepší. Konečně třetí možností je připustit i taková řešení, která povedou k dočasnému zhoršení účelové funkce (Lenstra, 2003). Simulované žíhání využívá k úniku z lokálního optima principu ochlazování taveniny a může být zapsáno pomocí následujícího pseudokódu:

Simulované žíhání

Start

Urči (k, T_k, y^i, V)

Opakuj

Pro $v = 1$ do V

Start

Generuj nový stav y^j v okolí y^i

$$F_{ij} = F(y^j) - F(y^i)$$

Když $F_{ij} \leq 0$, tak $y^j = y^i$ jinak

Když $F_{ij} > 0$ a $P(\Delta F) > \text{Rnd}(0,1)$,

tak $y^j = y^i$

Konec

$k = k + 1$

Aktualizuj teplotu žíhání T_k

Dokud není $T_k \leq T_f$

Konec

Zvolíme-li nějaký počáteční stav y^i oceněný hodnotou účelové funkce $F(y^i)$ a teplotu tohoto stavu ve výši T_k , pak nově generovaný stav oceněný hodnotou účelové funkce $F(y^j)$ bude přijat vždy, když $y^j \leq y^i$. V opačném případě bude nový stav přijat s pravděpodobností vyplývající z Boltzmannova distribučního zákona ve tvaru:

$$P(\Delta F) = \exp\left(\frac{F^i - F^j}{K_B T_K}\right) \quad (4.24)$$

kde K_B je Boltzmannova konstanta a T_k aktuální teplota. Na počátku, při vysoké teplotě, se tato pravděpodobnost blíží jedné, což má za následek snadné přijetí energeticky náročnějších stavů. Pravděpodobnost přechodu do vyššího energetického stavu ovšem klesá s každým snížením teploty žhání. Činnost algoritmu se zastaví, když aktuální teplota T_k poklesne pod zvolenou konečnou teplotu T_f (Osman and Kelly, 2012). Efektivnost metody je závislá na výběru následujících strategií a řídicích parametrů:

- strategie ochlazování a stanovení počáteční teploty,
- strategie generování sousedních stavů,
- kritérium ukončení výpočtu ve vnější a vnitřní smyčce algoritmu.

Strategie snižování teploty ve vnější smyčce algoritmu a určení počáteční teploty T_0 má zásadní vliv na efektivnost výpočtu. Počáteční teplotu lze získat například pomocí následující rovnice:

$$T_0 = \frac{-\Delta F_{ij, \max}}{\ln 0,9} \quad (4.25)$$

kde $K_B = 1$ a $\Delta F_{ij, \max}$ je předpokládaná maximální změna hodnoty účelové funkce při přechodu mezi jednotlivými stavy, kterou lze získat vyhodnocením dostatečného počtu náhodně generovaných stavů před zahájením běhu algoritmu. Nejčastěji používané strategie ochlazování jsou tyto (Van Laarhoven and Aarts, 1987):

1. Lineární ochlazování

Teplota žhání se v každém kroku sníží o hodnotu Δ , kde $\Delta = (T_0 - T_f)/R$. Konstantu R volíme jako kompromis mezi přesností (R velké) a rychlostí výpočtu (R malé).

2. Exponenciální ochlazování

Teplota žhání se v každém kroku vynásobí konstantou, jejíž hodnota se pohybuje mezi 0 (rychlé snižování teploty) a 1 (pomalé snižování teploty).

3. Adaptivní ochlazování

Teplota žíhání se snižuje na základě statistického vyhodnocení fluktuací účelové funkce na dané teplotní hladině. K výpočtu se nejčastěji využívá Aarts-van Laarhovenovy rovnice.

Strategie generování sousedních stavů je závislá na typu úlohy. U úloh permutačního typu (např. rozvrhování víceproduktových výrob) lze například využít metodu přesunu jednoho prvku, metodu přeuspořádání úseku či metodu vzájemné výměny obecně n prvků. Další možností, kterou lze uplatnit při optimalizaci (s, S) modelu řízení zásob, je zvyšování či snižování hodnot jednoho parametru o jedničku a generování hodnot ostatních parametrů pomocí náhodných čísel. Ukončení výpočtu ve vnější smyčce by mělo nastat v případě, že teplota žíhání T_k poklesne pod předem určenou konečnou teplotu T_f , kterou lze získat jako minimální změnu hodnoty účelové funkce $\Delta F_{ij, \min}$ vyhodnocením dostatečného počtu náhodně generovaných stavů před zahájením běhu algoritmu. Generování nových stavů ve vnitřní smyčce je na každé teplotní hladině praktické ukončit po V krocích, kde V je obvykle počet možných sousedních stavů aktuálního přijatého řešení (Van Laarhoven and Aarts, 1987).

V literatuře je zmíněno využití simulovaného žíhání například při řešení problému obchodního cestujícího (Ghosh, et al., 1988) či při optimalizaci pořadí a velikosti výrobních dávek (Koulamas, et al., 1994).

4.1.2.10 Tabu search

Tabu search heuristika využívá stejně jako metoda simulovaného žíhání algoritmu lokální optimalizace. Prohledávání je zahájeno v počátečního stavu y^i a pomocí vhodné procedury je generováno řešení y^j , které s počátečním stavem sousedí. Následuje vyhodnocení změny účelové funkce F_{ij} odpovídající přechodu systému ze stavu i do stavu j . Dojde-li k poklesu účelové funkce (v případě hledání minima), přepíše se výchozí stav y^i řešením y^j . Až potud je postup naprosto shodný s algoritmem lokální optimalizace, z čehož vyplývá, že prohledávání uvázne v lokálním optimu $F_{lok-min}$. V okamžiku, kdy je nutné připustit dočasné zhoršení účelové funkce, aby byl algoritmus vyveden z lokálního optima, se uplatní tzv. tabu seznam T . Do tohoto seznamu se postupně ukládají všechny použité transformace t v inverzním tvaru t^{-1} . Tabu seznam má ovšem omezenou délku, takže nejstarší vložená transformace je při vložení nové použité transformace ze seznamu vypuštěna. Transformace vložené do seznamu se nesmí v dalším průběhu používat, protože jejich aplikace by mohla vést algoritmus zpět do lokálního optima. Použití tabu seznamu způsobí, že při dosažení lokálního extrému bude algoritmus vychýlen a bude muset pokračovat v prohledávání nových oblastí. Vychýlení algoritmu se obvykle provádí náhodným generováním několika nových stavů, které algoritmus nasměrují do dosud neprohledaných oblastí. Prohledávání je nejčastěji ukončeno po uplynutí předem definované doby T_{max} či po překročení určitého počtu iterací, které nevedou ke zlepšení účelové funkce. Nalezené nejlepší řešení F_{min} se prohlásí za optimum (Glover and Laguna, 1999).

Uvedený postup je pro názornost zapsán v podobě pseudokódu níže.

Tabu search

Start

Urči $(T_{\max}, y^i); F_{\min} = \infty$

Opakuj

$F_{\text{lok-min}} = \infty$

pro všechna y^j z okolí y^i

Start

Použij transformaci $t: y^i \rightarrow y^j$

Spočti $F(y^j)$

Když $F(y^j) < F_{\text{lok-min}}$ tak

Start

$y^* = y^j, t^* = t^{-1}, F_{\text{lok-min}} = F(y^j)$

Konec

Konec

Když $F_{\text{lok-min}} < F_{\min}$ tak

Start

$F_{\min} = F_{\text{lok-min}}, y_{\min} = y^*$

Konec

$y^i = y^*$

Vlož transformaci y^* do seznamu T

Dokud není překročen čas T_{\max}

Konec

Efektivnost použití metody ovlivňují zejména následující faktory (Glover and Laguna, 1999):

- délka a typ tabu seznamu,
- velikost prohledávaného okolí,
- kvalita počátečního řešení.

Tabu seznam představuje v tabu search heuristice prvek krátkodobé paměti a umožňuje algoritmu rozpoznat, které oblasti již byly jednou navštíveny. Jeho délka je zásadní pro efektivnost výpočtu. Příliš krátký seznam bude mít za následek zacyklení algoritmu na nějaké opakující se sekvenci transformací, příliš dlouhý seznam naopak způsobí časté vynechávání nadějných bodů (Glover and Laguna, 1999).

Při použití tabu search heuristiky se obvykle využívají tři typy tabu seznamů. Nejpoužívanějším je pevný seznam, jehož nevýhodou je možnost zacyklení na nějaké sekvenci transformací. K překonání tohoto problému byl proto vyvinut tzv. dynamický seznam či strategie známá také jako pohyblivá clona, která zajistí, že při výpočtu je aktivní vždy pouze část seznamu (Hubscher and Glover, 1994).

Aplikace heuristiky tabu search v modelování a optimalizaci dodavatelských systémů jsou uvedeny např. v Lee and Kwon (2010), Griffis, et al. (2012), Jin, et al. (2014), Validi, et al. (2015) či Mirmajlesi and Shafaei (2016).

4.1.3 Analytické modely

Mezi analytické modely patří zejména teorie her a analytické metody založené na bázi vícekritériálního rozhodování mezi které patří data envelopment analysis, analytical hierarchy process, analytical network process a life cycle analysis [viz například Cook and Seiford (2009), Haq and Kannan (2006), Agarwal, et al. (2006)].

4.1.3.1 Teorie her

Pro teorii her je typická práce s předpokladem, že řešení nějaké rozhodovací situace neovlivňuje jediný individuální nebo kolektivní subjekt rozhodování. To do značné míry odpovídá většině rozhodnutí, která přijímají vedoucí pracovníci v tržních podmínkách. Tato rozhodnutí jsou totiž výrazně ovlivňována také rozhodnutími, která učiní další subjekty, a to zejména konkurenti na trhu. Jestliže konkrétní společnost přijme určitou cenovou strategii, ovlivní tím zajisté nejen vlastní výnosy, ale i výnosy konkurentů. To samozřejmě platí i naopak za předpokladu, že základním cílem podnikatelských subjektů je maximalizace výnosů. Dalším charakteristickým rysem rozhodovacích situací je skutečnost, že co jeden podnikatelský subjekt získá, to jiný ztratí. Jedná se tedy o princip antagonismu. Množina konkurentů na trhu je někdy označována jako racionální subjekty, což znamená, že se snaží racionálně dosáhnout nějakého žádoucího efektu. Na výsledky rozhodnutí ovšem působí i náhodné vlivy v podobě indiferentních účastníků, kteří nespojují přijatá rozhodnutí s nějakým efektem. Předmětem teorie her je modelování situací, u nichž je výsledek rozhodovacího procesu ovlivňován více účastníky, kteří buď mají zájem na výsledcích rozhodnutí, nebo sice výsledek rozhodnutí ovlivňují, ovšem bez zájmu o výsledný efekt. Základní pojmy teorie her jsou (Dyntar and Gros, 2015):

- hra – rozhodovací situace,
- hráči – účastníci rozhodovací situace,
- strategie – varianty rozhodnutí, které mohou hráči přijmout,
- bod hry – výsledek formulovaný přijatými strategiemi hráčů,
- výhra – přínos, který hráči získají.

Hry je možné na základě množiny strategií klasifikovat jako:

- konečné hry – množina strategií, které mohou hráči přijímat, je konečná,
- nekonečné hry – hráči mohou přijímat nekonečně mnoho strategií.

Hry je možné na základě celkové sumy výher všech hráčů klasifikovat jako:

- hry s konstantním součtem výher – suma výher, které hráči získají, je konstantní,
- hry s nulovým součtem výher – suma výher, které hráči získají, je nulová,
- hry s nekonstantním součtem výher – suma výher, které hráči získají, se mění.

Pro hry s konstantním součtem výher dále platí, že jde o antagonistické hry, kdy účastníci hry jsou vzájemně v přímém rozporu a vyšší výhra jednoho znamená pokles výhry ostatních hráčů. U her s nulovým součtem výher a s proměnným součtem výher hovoříme o neantagonistických hrách.

Pokud se hráči dohodnou na společném postupu, jedná se o hry kooperativní, v opačném případě hovoříme o hrách nekooperativních. Pokud na výsledky rozhodnutí nemají vliv indiferentní účastníci, jde o hry v normálním tvaru. Je-li možné pracovat při hledání optimálních strategií s maticí výher pouze jednoho racionálního hráče, jedná se o jednomaticové hry.

A konečně je možné hry klasifikovat na základě způsobu výběru optimálních strategií jako:

- hry, které mají řešení v čistých strategiích – hráč volí vždy pouze jednu strategii,
- hry, které mají řešení ve smíšených strategiích – hráč střídá všechny strategie s určitou pravděpodobností.

Aplikace teorie her v modelování a optimalizaci dodavatelských systémů jsou uvedeny např. v Audy, et al. (2012), Zamarripa, et al. (2012), Yin, et al. (2015), Soleimani, et al. (2016) či Gao and You (2017).

4.1.3.2 Data envelopment analysis

Data envelopment analysis je metoda, která na základě formulace a řešení úlohy matematického programování umožňuje srovnání efektivity prvků dodavatelského systému v situacích, kdy velké množství různorodých vstupů a výstupů činí tuto komparaci obtížnou (Zhang, et al., 2014). Prvky dodavatelského systému mohou být například maloobchody, výrobní podniky, podnikové divize či sklady distributorů. Klasické míry stanovení efektivity nějakého prvku dodavatelského systému se opírají o stanovení podílu výstupů, které prvek vytvoří a spotřeby vstupů využívaných prvkem k jeho činnosti. V případě skladu distributora je vstupem například zásoba či osobní náklady pracovníků skladu a

výstupem počet vyřízených objednávek maloobchodu či úroveň služeb poskytovaná zákazníkům v podobě termínu vyřízení objednávky a jeho rozptylu. Farrell and Fieldhouse (1962) navrhli využít ke stanovení efektivity obecně *j-tého* prvku dodavatelského systému následujícího ukazatele:

$$Efektivita_j = \frac{\sum_{i=1}^m u_i y_i}{\sum_{k=1}^n v_k x_k} \quad (4.26)$$

kde u_i je váha *i-tého* výstupu *j-tého* prvku, y_i je množství *i-tého* výstupu, které tento prvek vytvoří, v_k je váha *k-tého* vstupu *j-tého* prvku a x_k je množství *k-tého* vstupu, které tento prvek spotřebuje. Problémem ukazatele v této podobě je potřeba stanovení vah, které je v reálných situacích zkresleno subjektivitou rozhodovatele snažícího se o dosažení lepšího výsledku, než je skutečnost, a často je také otázka, jakým způsobem měřit množství spotřebovaných vstupů a vytvořených výstupů. Charnes, et al. (1978) navrhli doplnit stanovení efektivity o soustavu omezení tak, aby došlo k odstranění zkreslení vah vstupů a výstupů způsobeného subjektivním pohledem rozhodovatele. Úloha stanovení efektivity má následující tvar:

$$\text{Max } Efektivita_j = \frac{\sum_{i=1}^m u_i y_i}{\sum_{k=1}^n v_k x_k} \quad (4.27)$$

$$\frac{\sum_{i=1}^m u_i y_i}{\sum_{k=1}^n v_k x_k} \leq 1 \text{ pro každé } j \quad (4.28)$$

$$Efektivita_j \leq 1 \quad (4.29)$$

$$u_i, v_k \geq \varepsilon \quad (4.30)$$

Výše popsaná úloha je úlohou nelineárního programování. Je snadno řešitelná linearizací a následným využitím simplexové metody. Proměnné úlohy jsou váhy vstupů a výstupů, jejichž hodnota je vyšší než malé kladné číslo ε . Je to z toho důvodu, aby nějaký vstup a výstup nebyl při výpočtu efektivity ignorován. Hodnota efektivity rovná 1 znamená, že *j-tý* prvek dodavatelského systému je v porovnání s ostatními efektivní, hodnota efektivity menší než 1 znamená, že i při zvolení optimálních vah (jde o úlohu maximalizace efektivity) je nějaký prvek efektivnější, přičemž argument, že výpočet efektivity je ovlivněn subjektivním stanovením vah, je neplatný. Odstranění subjektivity ze stanovení vah při výpočtu efektivity je ovšem pouze relativní v případě, že prvek dodavatelského systému vykazuje nějaký výstup a vstup, a že je mezi nimi výrazný nepoměr ve prospěch výstupu. To totiž znamená, že tomuto poměru je přiřazena nejvyšší váha, a jde-li u každého porovnávaného prvku dodavatelského systému o různou dvojici výstup/vstup, jeví se všechny prvky jako efektivní. Bude-li například efektivita prvků dodavatelského systému hodnocena 5 vstupy a 5 výstupy, existuje reálná možnost až 25 efektivních jednotek. Z tohoto důvodu se v literatuře objevují postupy, jak omezit hodnoty vah tak, aby bylo

dosaženo kompromisu mezi subjektivitou rozhodování prvků dodavatelských systémů a poněkud diskriminačním charakterem data envelopment analysis [viz např. Dyson and Thanassoulis (1988), Doyle and Green (1994) či Dyson, et al. (2001)].

Aplikace data envelopment analysis v modelování a optimalizaci dodavatelských systémů jsou uvedeny např. v Dotoli, et al. (2010), Grigoroudis, et al. (2014), Dotoli, et al. (2015), Che and Chang (2016) či Babazadeh, et al. (2017).

4.1.3.3 Analytic hierarchy process

Analytic hierarchy process je modelová technika využívaná k analýze komplexních rozhodovacích problémů. Její základy popsal Thomas Saaty v roce 1988 [viz Saaty (1988)]. Pro analytic hierarchy process je charakteristické spojení matematických modelů a psychologie, které rozhodovateli napomáhá hlouběji pochopit řešený problém a na základě toho najít takové řešení, které nejlépe odpovídá jeho cíli. Základem řešení je dekompozice problému do hierarchické struktury tvořené subproblémy, z nichž každý může být řešen nezávisle (Saaty, 2008). Hierarchie představuje systém složený z vrstev, který umožňuje řadit a organizovat hmotné a nehmotné zdroje a jejich charakteristiky (tj. prvky) tak, že každý prvek systému patřící do určité vrstvy (kromě prvku vrcholového) je zároveň nadřazený a podřazený nějakým prvkům ve vrstvách ostatních. Nadřazenost a podřazenost prvků je nejčastěji znázorňována pomocí pyramidových diagramů, ale lze ji popsat také matematicky. Mezi prvky, které tvoří hierarchický systém, patří cíl, jehož má být dosaženo (tj. vrcholový prvek), alternativy, které vedou k dosažení cíle, a faktory (kritéria), které tyto alternativy popisují. Kritérium nabývá určitých hodnot vyjádřených slovně či číselně, přičemž hodnoty kritéria vyjadřují žádoucí či nežádoucí charakteristiku alternativy vzhledem k naplnění cíle. K zobrazení cíle, alternativ a kritérií se v diagramech obvykle využívá obdélníků (tj. uzlů) a jejich vzájemné propojení se zakresluje pomocí hran. Po zakreslení diagramu následuje párové srovnání cíle s jednotlivými kritérii s cílem odhadnout, do jaké míry kritéria přispějí k dosažení cíle. Tento krok se nazývá stanovení důležitosti kritérií. Dále jsou párově porovnávána všechna kritéria se všemi alternativami s cílem stanovit jejich preference. Číselné vyjádření důležitosti kritérií a preferencí alternativ se nazývá priorita uzlu. Priorita uzlu je číslo mezi 0 a 1 a vyjadřuje relativní váhu uzlu ve skupině uzlů (tj. ve vrstvě). Například pro dva uzly ze stejné skupiny, které mají priority 0,2 a 0,1, platí, že uzel s prioritou 0,2 má na dosažení cíle dvakrát větší vliv než uzel s prioritou 0,1. Konečné rozhodnutí pak spočívá v sestupném seřazení alternativ dle priorit a výběru alternativy s prioritou nejvyšší. Z výše uvedeného popisu vyplývá, že analytic hierarchy process se s úspěchem využívá při řešení následujících problémů:

- výběr nejlepší alternativy ze skupiny alternativ v případě rozhodování na základě více kritérií,
- seřazení alternativ od nejlepší po nejhorší vzhledem k vlivu alternativ na dosažení cíle,

- stanovení vzájemné relativní váhy alternativ, vyjadřující vliv výběru alternativy na dosažení cíle,
- rozdělení zdrojů mezi alternativy na základě jejich priorit,
- porovnání výkonosti procesů v podniku s podnikem, který dosahuje nejlepších výsledků (tj. benchmarking),
- řešení konfliktních rozhodovacích situací,
- zlepšování kvality.

Aplikace konceptu analytic hierarchy process v modelování a optimalizaci dodavatelských systémů jsou uvedeny např. v UmaDevi, et al. (2012), Ertugrul and Ozbay (2013), Ozgen and Gulsun (2014), Jakhar (2015) či Entezaminia, et al. (2016).

4.1.3.4 Analytic network process

Analytic network process vychází z principů analytic hierarchy process s tím rozdílem, že cíl, kritéria a alternativy jsou při popisu rozhodovacího problému rozšířeny do podoby sítě. Reprezentace rozhodovacího problému pomocí sítě je výhodná v situacích, kdy mezi skupinami kritérií a alternativ ve vrstvách existují závislosti [viz Saaty (1996) či Saaty (2005)]. Například při nákupu auta na základě ceny, výkonu a bezpečnosti je zřejmé, že existuje vzájemná závislost jak mezi cenou a bezpečností, tak mezi cenou a výkonem. Postup metody je jinak velmi podobný konceptu analytic hierarchy process a spočívá v párovém srovnání cíle s jednotlivými kritérii a párovém srovnání kritérií s alternativami. Konečné rozhodnutí je založeno na výběru alternativy s nejvyšší prioritou.

Aplikace konceptu analytic network process v modelování a optimalizaci dodavatelských systémů jsou uvedeny např. v Agarwal and Vijayvargy (2012), Ashlaghi (2014) či Gandhi, et al. (2016).

4.1.3.5 Life cycle analysis

Life cycle analysis je koncept, který posuzuje životní cyklus výrobků a služeb z hlediska jejich dopadu na životní prostředí. Cílem konceptu je stanovit, jakým způsobem realizace materiálových toků spojených s dodávkami služeb a výrobků zákazníkům zatěžuje životní prostředí, a to včetně zpětných toků (In and Curran, 1994). Životní cyklus v klasickém pojetí je rozdělen na výrobu materiálů, výrobu hotových výrobků a jejich distribuci. V tomto pojetí hovoříme o lineárním dodavatelském systému či o dodavatelském systému cradle to grave (Barton, et al., 1996). Naproti tomu life cycle analysis zdůrazňuje s ohledem na požadovanou minimalizaci dopadů aktivit spojených s dodávkami výrobků a služeb na životní prostředí, potřebu přechodu na cirkulární ekonomiku, kdy jsou materiálové toky realizovány v cradle to cradle dodavatelských systémech (Braungart, et al., 2007). Tyto dodavatelské systémy pracují s dvojitou logikou cirkulace materiálových toků. V případě látek organického původu, které jsou snadno odbouratelné, je z hlediska životního prostředí

možné vracet je zpět do biosféry (princip lineárního dodavatelského systému). Látky syntetické musí být do produktů vkládány tak, aby bylo možné je z nich následně extrahovat a opět použít (princip cirkulace).

Life cycle analysis se opírá o:

- sestavení soupisu energetických a materiálových vstupů pro každý výrobek a jejich úniků do životního prostředí,
- vyhodnocení potenciálních dopadů spojených s identifikovanými vstupy a úniky,
- interpretaci výsledků za účelem získání informací pro podporu rozhodování.

V souladu s terminologií, která je součástí norem ISO 14040 a 14044, lze výše uvedené kroky dekomponovat do čtyř fází, kterými jsou:

- stanovení cíle a rozsahu,
- inventarizační analýza,
- posuzování dopadů životního cyklu,
- interpretace výsledků.

V rámci fáze stanovení cíle dochází k vymezení důvodu provádění metody, zamýšleného použití a také k definování toho, pro koho se studie zpracovává a kdo bude s jejími výsledky seznámen. Normou je dále stanoven i rozsah studie, přičemž je největší pozornost věnována především stanovení hranic systému, funkční jednotky a požadavkům na kvalitu údajů. Platí, že hranice systému musí být vybrány v souladu se stanoveným cílem studie, a že tyto hranice jsou určeny jednotkovými procesy (Remtová, 2003). Velkou pozornost je třeba věnovat stanovení funkční jednotky, neboť ta představuje bázi, ke které se bude vztahovat negativní dopad na životní prostředí. Jednotka proto musí být jasně definována a musí být měřitelná. Správnost stanovení funkční jednotky je důležitá i při porovnání negativních vlivů produktových systémů, které v praxi plní tutéž funkci.

Inventarizační analýza vyžaduje podrobnou znalost výrobních operací, jejich parametrů, vlivů na životní prostředí a přesného materiálového složení všech surovin použitých při výrobě. Uvedená analýza spočívá ve sběru údajů a stanovení odpovídajících vstupů a výstupů ze systému. Zahrnuje především sběr primárních dat a posouzení jejich věrohodnosti, reprodukovatelnosti a transparentnosti. Výstup analýzy má podobu inventarizační matice. V matici jsou ve sloupcích uvedeny jednotlivé etapy životního cyklu výrobku a v řádcích vlivy na životní prostředí.

Posuzování dopadů životního cyklu se zaměřuje na hodnocení významu potenciálních environmentálních dopadů. Skládá se z:

- výběru kategorií dopadu, indikátorů kategorie a charakterizačních modelů,

- klasifikace,
- charakterizace,
- vyhodnocení.

Klasifikace spočívá v třídění všech negativních vlivů uvedených v inventarizační matici do kategorií podle jejich působení na životní prostředí. Emise (výstupy) jsou nejčastěji tříděny podle efektů druhého řádu. Tyto efekty se nejčastěji dělí na globální (skleníkový efekt a narušování ozónové vrstvy), regionální (acidifikace, eutrofizace, tvorba smogu a ekotoxicita) a lokální (akutní toxické působení na člověka, kontaminace půd, zápach a hluk). Charakterizace je posouzením celkového dopadu na životní prostředí z kvantitativního hlediska. Výsledkem je standardizovaný profil výrobku.

Interpretace výsledků life cycle analysis spočívá ve stanovení relativního významu všech získaných dílčích zátěží. V této fázi je nejdůležitější především zachování transparentnosti postupu.

Hlavní přínosy life cycle analysis jsou (Kočí, 2010):

- vzájemné srovnání environmentálních dopadů produktů s ohledem na jejich funkci,
- hodnocení environmentálních dopadů s ohledem na celý životní cyklus produktu,
- zavedení hranic systému pro jasné vyjádření rozsahu produktového systému,
- vyjadřování zásahů do životního prostředí nikoli výčtem emisních toků, ale použitím definovaných kategorií dopadu,
- převedení hmotnostně vyjádřených emisních toků na konkrétní hodnoty výsledků indikátorů kategorií dopadu,
- schopnost identifikovat přenášení environmentálních problémů jak v prostoru, tak mezi různými kategoriemi dopadu.

Aplikace konceptu life cycle analysis v modelování a optimalizaci dodavatelských systémů jsou uvedeny např. v de Albuquerque, et al. (2013), Vaskan, et al. (2014), Gonela, et al. (2015), Cadavid-Giraldo, et al. (2016) či Lieder, et al. (2017).

4.2 Supply chain operations reference model

Supply chain operations reference (SCOR) model je referenční model určený primárně pro standardizovaný popis dodavatelských systémů (Thunberg and Persson, 2013). V současné

době je udržován a aktualizován neziskovou organizací Supply chain council. SCOR model obsahuje 3 základní části, kterými jsou:¹

- soubor předdefinovaných procesů probíhajících v dodavatelském systému,
- soubor metrik používaných k posouzení výkonu dodavatelského systému a
- doporučené nejlepší přístupy k návrhu a řízení procesů v dodavatelském systému.

Mezi procesy probíhající v dodavatelském systému se řadí:

- plánování (Plan),
- získávání zdrojů potřebných pro činnost (Source),
- výroba (Make),
- dodání zboží či služby (Deliver),
- realizace zpětných toků (Return).

Tyto základní procesy tvoří nejvyšší, první stupeň modelu a jsou ve druhém stupni detailněji popsány pro:

- výrobu na sklad,
- výrobu na zakázku,
- design a výrobu na zakázku.

Ve třetím stupni jsou veškeré procesy z druhého stupně rozloženy na činnosti a detailně popsány.

Soubor metrik používaných k posouzení výkonu dodavatelského systému je rovněž rozdělen do 3 stupňů. Tyto metriky slouží ke srovnání dodavatelských systémů v rámci odvětví, neboť Supply chain council udržuje a poskytuje aktualizovanou databázi těchto metrik svým členům. Struktura metrik je standardizována a je tvořena tak, že metriky z nejvyššího stupně jsou agregované ukazatele tvořené metrikami z nižších stupňů.

SCOR model v podobě nástroje pro reengineering procesů probíhajících v dodavatelském systému je nejčastěji používán k popisu a mapování současného stavu, identifikaci potenciálu zlepšení a návrhu stavu budoucího dle požadavků uživatele. K tomu

¹ <http://www.apics.org/sites/apics-supply-chain-council/frameworks/scor>; citováno 24. 11. 2017.

slouží databáze doporučených nejlepších přístupů k návrhu a řízení procesů, které svým členům poskytuje rovněž Supply chain council. SCOR model sám o sobě neobsahuje žádný modelový přístup k návrhu a optimalizaci dodavatelského systému, a proto je velmi často kombinován s metodami popsány v předchozí kapitole. SCOR model byl úspěšně aplikován v různých oborech. Uvedme si nyní některé jeho aplikace popsané v literatuře:

Tabulka 4.1 Aplikace SCOR modelu v modelování reálných dodavatelských systémů

Publikace	Stručný popis aplikace	Odvětví
Röder and Tibken (2006)	Návrh nástroje pro podporu rozhodování o struktuře dodavatelského systému.	Automobilový průmysl
Burgess and Singh (2006)	Hodnocení vlivu sociálních a politických aspektů na výkon dodavatelského systému.	Hutnictví
Han and Chu (2009)	Hodnocení efektivity dodavatelského systému při výrobě elektrické energie.	Energetika
Persson and Araldi (2009)	Posouzení vlivu kvality vstupních materiálů na výkon dodavatelského systému.	Automobilový průmysl
Banomyong and Supatn (2011)	Test věrohodnosti SCOR modelu při hodnocení výkonu dodavatelského systému.	Různá

4.3 Simulační modelování

Simulační metody se opírají o formulaci logicko-matematického modelu a jeho vytvoření na počítači. Pomocí experimentů s počítačovým modelem lze studovat modelovaný systém a získat odpovědi na otázky typu „co se stane, když...“. Mezi simulační metody patří zejména spreadsheet simulace, system dynamics a discrete event simulace.

Dynamická simulace představuje vynikající nástroj pro plánování, návrh, optimalizaci a reengineering dodavatelských systémů. O významu, který je tomuto nástroji modelování přikládán, svědčí některé vybrané charakteristiky:

Pomocí počítačového modelu, „který věrně zachycuje skutečné podmínky a specifiky ovlivňující celkovou výkonnost a efektivnost modelovaného objektu, lze nasimulovat reálnou či plánovanou funkci daného systému dle požadavků uživatele. Možnosti využití simulace jsou nesmírně široké, jedná se o přístup, který svojí univerzálností vysoce překonává klasické statické matematické modely.“ (Glover, et al., 1999)

Simulace je proces tvorby logicko-matematického modelu reálného objektu, systému na něm definovaného nebo procesu rozhodování a realizace velkého množství experimentů s ním, jejichž cílem je (Gros, 1996):

- popis systému,

- poznání jeho funkce,
- odhad jeho budoucího chování,
- nalezení řešení problému, který mnohdy ústí do
- návrhu a ověření funkce nové struktury systému.

Simulace je metodou, která umožňuje pomocí počítačového modelu podnikového procesu předvídat chování systému při změně vnitřních či vnějších podmínek, optimalizovat podnikové procesy vzhledem k zadaným kritériím (zisk, náklady, spolehlivost) při současném respektování omezujících podmínek a porovnat mezi sebou navrhované alternativy organizace studovaného procesu (Dlouhý, et al., 2007). Použití simulačního modelu umožňuje provádět experimenty bez rizika negativního dopadu na reálný objekt.

Simulační modelování můžeme také charakterizovat jako proces tvorby a experimentování s matematickým modelem fyzického systému převedeným do počítače (Chung, 2003). Cílem experimentování je nalezení takových hodnot výstupních veličin modelu, které vyhovují předem stanoveným požadavkům (Manlig, 1999).

4.3.1 Využití simulace

Simulační modelování a analýza se používají k následujícím účelům:

1. Získání náhledu na operace v systému

Některé systémy jsou natolik komplexní, že je obtížné pochopit veškeré vztahy uvnitř systému bez použití dynamického modelu. Jinými slovy, je nemožné studovat systém pomocí toho, že jej zastavíme a budeme izolovaně sledovat některé jeho komponenty (Chung, 2003). Simulační model tuto možnost poskytuje.

2. Rozvíjení postupů pro efektivní řízení operací a zdrojů

Můžeme mít systém, kterému rozumíme, ale jehož fungování chceme vylepšit. Dva základní přístupy tkví ve změně řízení operací, nebo řízení zdrojů. Změny v řízení operací mohou zahrnovat různé pořadí výrobních dávek, změny v řízení zdrojů, změny v přiřazování pracovníků nebo plánování odstávek apod. (Banks, 1998).

3. Testování nových koncepcí a systémů před jejich implementací

Pokud systém dosud neexistuje nebo pokud zvažujeme pořízení nového systému, simulační modely nám mohou poskytnout informaci o tom, jak budou nabízené

systémy pracovat (Kelton and Law, 2000). Náklady na simulaci jsou nízké v porovnání s náklady na kapitálové investice do velkých výrobních celků a zařízení. Simulace nám může také pomoci nastavit parametry pořizovaného zařízení.

4. Získání informací bez ovlivnění systému

Simulační modely jsou jediným uplatnitelným přístupem u systémů, které nesmí být ovlivněny. Typickým příkladem může být bezpečnostní kontrola na letišti, kde snížení počtu pracovníků může mít dopady na kvalitu práce i na její efektivitu (Chung, 2003).

S rostoucím výkonem výpočetní techniky a rozšířením simulačních programů, které umožňují vizualizaci a stavbu modelu v grafickém prostředí, se stává simulace stále rozšířenější i mezi managementem. Simulace přestává být pouze prostředkem pro použití ad hoc při reengineeringu procesů, stále častěji se stává prostředkem pro podporu rozhodování na pravidelné bázi.

4.3.2 Výhody simulace

Kelton and Law (2000) popisují následující výhody simulace:

1. Správný výběr variant

Simulace umožňuje vyzkoušet předem všechny navrhované varianty bez nutnosti vynaložit zdroje na jejich realizaci. Je tudíž možné vybrat nejlepší variantu uspořádání procesu ještě před vynaložením prostředků na jeho realizaci, což je výhodné v případech velkých investic či v případě existence rizika spojeného s rozhodnutím.

2. Zrychlení a zpomalení času

Simulace umožňuje zrychlit nebo zpomalit čas v modelu. Můžeme tak urychlit studování činností, nebo naopak činnost zpomalit a studovat ji detailně. Zrychlování času pomocí simulace představuje výhodu zejména v situacích, kdy je třeba opakovaně prověřovat varianty uspořádání procesu, tj. například v operativním plánování.

3. Diagnóza problémů

Většina procesů ve výrobních podnicích je značně komplexní, a je tedy obtížné vést v patrnosti veškeré závislosti mezi jednotlivými částmi procesu. Proces tvorby simulačního modelu umožňuje lépe porozumět procesům a vztahům mezi

proměnnými modelu. Získáme tak přehled o vlivu proměnných na modelovaný systém a můžeme navrhnout jejich optimální nastavení.

Banks (1998) vidí největší výhody simulace v možnosti:

1. Pochopit proč

Pomocí simulace, pomocí rekonstrukce události a jejího podrobného zkoumání, můžeme zjistit, proč daný jev nastává. To si v reálném systému dovolit nemůžeme, protože je obtížné jej vidět v jeho celistvosti.

2. Prozkoumat možnosti

Pokud disponujeme validním modelem, můžeme na něm zkoušet nové postupy bez nutnosti zásahu do reálného systému. Model poskytuje informace o důsledcích navrhovaných změn, které často vedou k návrhu efektivnějšího uspořádání procesu.

3. Identifikovat omezení

Často se zapomíná, že úzká místa jsou spíše důsledkem než příčinou. Simulace pomáhá identifikovat příčiny vzniku úzkých míst a jiných omezení.

4. Vizualizace modelu

Dnešní simulační programy umožňují vytvořit model, na který je možné pohlížet v různém stupni zvětšení a z mnoha pohledů, mnohdy i v prostředí virtuální reality. To usnadňuje identifikaci případných chyb již v procesu přípravy plánů.

5. Příprava na budoucnost

V neposlední řadě umožňuje simulační model vytvářet scénáře, čímž pomáhá nalézt odpovědi na otázky, „co se stane když...“. Model umožňuje měnit parametry, které představují vnější vlivy na systém, např. snížení poptávky po zboží apod.

6. Uvážlivé investování

Typické náklady na simulaci jsou menší než 1 % z celkových nákladů na implementaci navrhovaného řešení. Protože změnit vlastnosti jednou vybudovaného zařízení je velmi nákladné (zejména v prostředí chemické výroby při investici do velkých výrobních celků), je vhodné použít simulaci pro ověření parametrů investice.

4.3.3 Nevýhody simulace

Ačkoliv simulace umožňuje řešit problémy, na které klasické metody nestačí, a přináší celou řadu pozitiv, je třeba upozornit na některé její nevýhody.

1. Tvorba modelu vyžaduje speciální znalosti.

Pro tvorbu simulačního modelu jsou nutné dvě základní kompetence, a to schopnost pracovat s daným simulačním nástrojem a schopnost detailně porozumět modelovanému procesu. Programovací jazyky používané v simulaci vyžadují jejich velmi dobrou znalost a jsou často přístupné jen specialistům programátorům (Gros, 1996). Současně však existují nástroje orientované graficky, které umožňují intuitivnější sestavení modelu pomocí grafických prvků a jejich napojení. Takové simulační programy zpřístupňují simulace širokému okruhu manažerů a pracovníků obeznámených s chováním reálného systému (Banks, 1998).

2. Výstupy simulace může být obtížné interpretovat.

Výstupy simulace jsou v drtivé většině případů založeny na náhodném chování vstupních proměnných, a proto je jejich chování také náhodné. Někdy je obtížné odlišit, jestli je variabilita výstupů otázkou náhody, nebo důsledkem vztahů v modelu. Tvůrci simulačních programů se proto snaží poskytovat nástroje pro analýzu výstupů modelu (Banks, 1998).

3. Simulace může být časově náročná a drahá.

Šetření ve fázi přípravy modelu a nedostatek dat může způsobit, že model nebude poskytovat očekávané výstupy. Nevhodně stanovený rozsah a podrobnost modelu může vést k vysoké časové náročnosti běhu simulace, přičemž přínos z detailního modelování může být mizivý (Banks, 1998).

4. Simulace může být nevhodná pro daný účel.

Simulace je někdy používána v případech, kdy to není nezbytné a kdy je dokonce vhodnější použít analytické metody. Příkladem mohou být problémy řešitelné pomocí teorie hromadné obsluhy (Winston and Goldberg, 2004).

5. Simulace nemůže poskytnout jednoduché odpovědi na složitý problém.

V případě modelování složitého systému je pravděpodobné, že simulace neposkytne jednoduché odpovědi a že výstupy budou stejně komplexní jako systém sám. Často je nutné učinit zjednodušující předpoklady zejména proto, aby

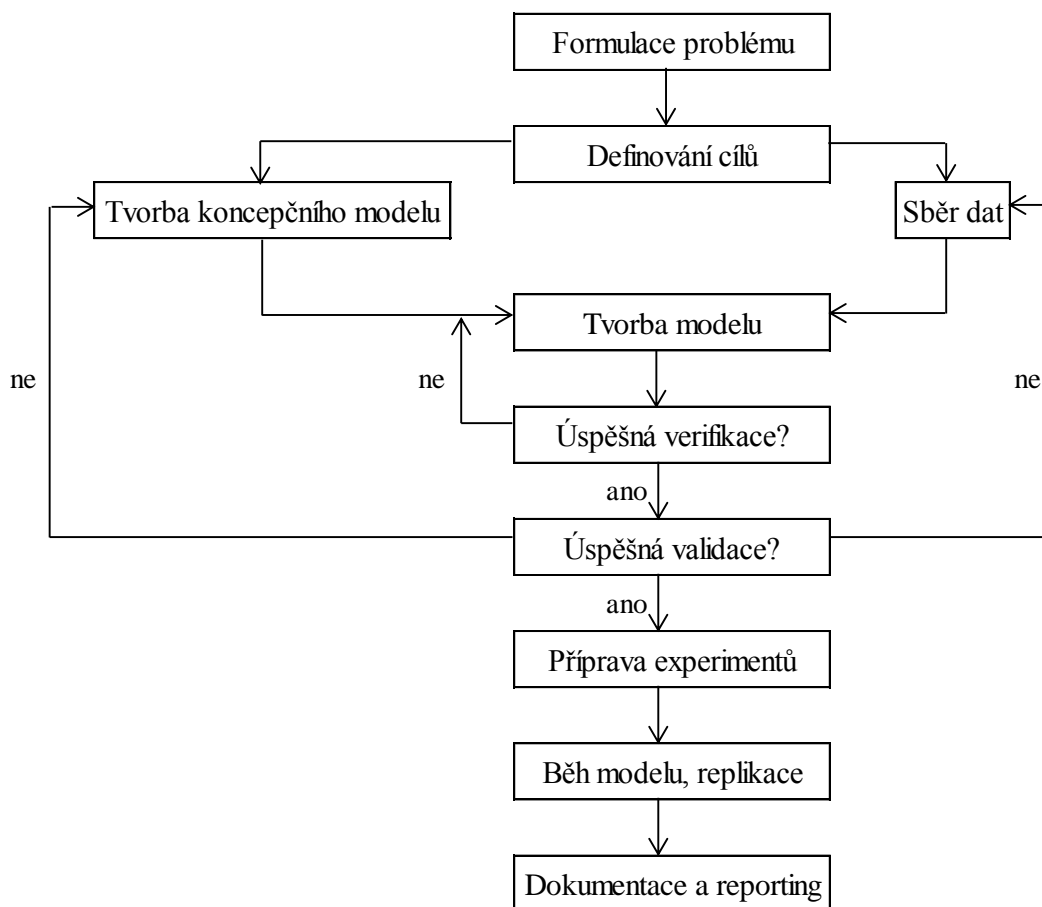
bylo možné model vytvořit v relativně krátkém čase. Pak ale hrozí riziko, že budou pomínuty některé kritické faktory systému a výsledné akce vzešlé ze simulačního modelu budou mít nízkou vypovídací hodnotu (Chung, 2003).

6. Simulace nemůže problém vyřešit.

Mnozí manažeři se domnívají, že simulace vyřeší jejich problémy. Simulace však může pouze pomoci vyhodnotit varianty a poskytnout podklady pro řešení problému. Samotné řešení problému je pak závislé na dobré implementaci závěrů simulace (Chung, 2003).

4.3.4 Tvorba simulačního modelu

Podle Dlouhý, et al. (2007) lze postup tvorby simulačního modelu rozdělit zhruba do šesti fází. Jednotlivé etapy jsou zobrazeny na Obr. 4.2, v následující části práce si je popíšeme.



Obr. 4.2 Etapy tvorby simulačního modelu

1. Definice problému a stanovení cíle

Prvním krokem při tvorbě modelu je stanovení cíle a definice problému. Zásadní je určit, zda je simulace vhodným prostředkem pro dosažení stanoveného cíle. Dobře formulovaný cíl usnadní provedení následujících etap.

2. Stanovení konceptu modelu a požadavků na data

Rozsah modelu musí být takový, aby umožnil dosáhnout stanoveného cíle. V této fázi je třeba utvořit si základní představu o modelovaném systému (tzv. konceptuální model) a odpovědět na následující otázky:

- Jaký podnikový systém modelujeme?
- Kdo jsou zákazníci systému?
- Podle jakých kritérií je hodnocena efektivnost systému?
- Jaká úroveň podrobnosti je nutná?
- Jaké objekty, činnosti a zdroje modelovaný systém zahrnuje?
- Jakým způsobem se přidělují omezené zdroje jednotlivým procesům?

3. Sběr dat

Simulace je metodou, jejíž vypovídající schopnost je silně ovlivněna kvalitou vstupních dat. I v situaci, kdy jsou data k dispozici, je nutné tato data ověřit. Například v situaci, kdy je čas určitých operací normován, je nutné zjistit, zdali to, co je vykazováno, odpovídá realitě.

Data použitá v modelu můžeme rozdělit na (Gros, 2009):

a. Vstupní proměnné

Tyto proměnné dělíme na proměnné říditelné, které jsou pod kontrolou řešitele a u nichž sledujeme jejich vliv na model, a na proměnné náhodné, které je nutné v průběhu simulace generovat za použití rozdělení pravděpodobností. Parametry těchto rozdělení jsou zjišťovány statisticky ze získaných souborů dat.

b. Stavové proměnné

Charakterizují v jednotlivých časových okamžicích stav modelovaného systému.

c. Parametry modelu

Jde o proměnné, které se v průběhu simulace nemění. Využívají se např. pro kvantifikaci kritérií pro posuzování variant struktury modelu.

d. Výstupní proměnné

Charakterizují výstupy modelu. Jde často o sérii ukazatelů, které byly simulací získány. Jedná se o využití strojů, využití pracovníků, doby čekání, délky front, průměrné stavy zásobníků, počty vyrobených kusů aj.

e. Základní vazby modelu

Jde o logické vazby mezi jednotlivými proměnnými a objekty v simulačním modelu. Může jít o reálná, v praxi pozorovatelná spojení (dopravníky, potrubí), ale také o pravidla (pořadí typů výrobku na lince).

4. Tvorba modelu

Vlastní tvorba modelu spočívá v převedení konceptuálního modelu do modelu v simulačním programu. Jde tedy o znázornění logiky reálného systému pomocí programovacího jazyka simulačního programu. Tvorba počítačového modelu je první kontrolou konceptuálního modelu, neboť nekompromisní přesnost počítačové logiky odhalí to, co bylo přehlédnuto při tvorbě konceptuálního modelu.

5. Verifikace a validace modelu

Verifikací rozumíme ověření toho, zda je vytvořený počítačový model v souladu s původním konceptuálním modelem. Validací modelu chápeme ověření toho, zda je počítačový model ve shodě s realitou. Model lze ověřit, pokud jsou k dispozici reálná data. Při validaci modelu lze tolerovat jistou míru nepřesnosti, neboť model je pouze zjednodušením reality a měl by být podrobný pouze do té míry, aby posloužil ke splnění cíle, který byl definován v první etapě.

6. Provedení experimentů a analýza výsledků

Experimentální fáze a fáze analýzy výsledků jsou často opomíjené, avšak důležité části práce s modelem. Během nich jsou prováděny plánované experimenty a zkoušeny připravené varianty, které jsou následně vyhodnoceny.

Vrcholem práce na modelu nemůže být jediná realizace simulačního běhu. Vyskytují-li se ve vstupních charakteristikách náhodné veličiny, jsou i výstupní charakteristiky náhodnými veličinami, takže výsledkem analýzy je buď bodový, nebo intervalový odhad. Proto je třeba dívat se na simulaci jako na počítačový statistický experiment.

4.3.5 Generování náhodných čísel v simulačních modelech

Vzhledem k tomu, že vstupní proměnné modelu mají často náhodný charakter, je nutné v průběhu simulace generovat jejich hodnoty z rozdělení pravděpodobností za použití vhodné metody. Základním zdrojem náhodných proměnných jsou proměnné r rovnoměrného rozdělení pravděpodobností s funkcí hustoty pravděpodobnosti $f(r)=1$ pro $0 < r < 1$ a $f(r)=0$ pro r ostatní a distribuční funkcí $F(r)=0$ pro $r \leq 0$, $F(r)=r$ pro $0 < r < 1$ a $F(r)=1$ pro r ostatní (Dyntar and Gros, 2015). Hodnoty náhodné proměnné rovnoměrného rozdělení pravděpodobností se pohybují v intervalu $(0,1)$, přičemž každá hodnota disponuje stejnou pravděpodobností svého výskytu. Zdrojem náhodných proměnných mohou být tabulky náhodných čísel, které lze najít ve většině statistických tabulek. Stejně tak je k dispozici generátor náhodných čísel rovnoměrného rozdělení například v MS Excel v podobě funkce NÁHČÍSLO() či v kódu programovacího jazyka Visual Basic for Applications v nástroji Makra v podobě funkce Rnd(). Generátory náhodných čísel rovnoměrného rozdělení jsou také někdy označovány jako generátory pseudonáhodných čísel. Je to z toho důvodu, že pro jejich generování se využívá metoda, která generuje dlouhé sekvence náhodných čísel, jež se po čase mohou opakovat. Většinou je využívána Lehmerova metoda [viz Payne, et al. (1969) či Park and Miller (1988)], které sice nezaručí úplnou nahodilost členů řady, ale možná periodicitu opakování náhodných čísel je velmi dlouhá a nemůže v reálných aplikacích ovlivnit systematicky dosažené výsledky. Lehmerova metoda generování náhodného čísla r_n je v prvním kroku založena na vztahu:

$$x_{n+1} = (a \cdot x_n + c) \cdot (\text{mod } m) \quad (4.31)$$

kde x_0 je počáteční hodnota posloupnosti, a a c jsou vhodně zvolená čísla, číslo m se nazývá modul a zápis $(\text{mod } m)$, tj. modulo m , značí zbytek po celočíselném dělení. Z úpravy této rovnice na tvar:

$$x_{n+1} = a \cdot x_n + c - m \cdot \left\lfloor \frac{a \cdot x_n + c}{m} \right\rfloor \quad (4.32)$$

kde výraz v hranaté závorce značí celou část čísla po dělení, vyplývá, že hodnoty posloupnosti jsou celočíselné zbytky po dělení číslem m a generované hodnoty náleží do množiny $(0, 1, \dots, m-1)$. Pokud mají ležet generované hodnoty náhodných čísel r_n v intervalu $(0,1)$ musí platit:

$$r_n = \frac{x_n}{m} \quad (4.33)$$

a v případě, že dojde ve vygenerované posloupnosti k výskytu hodnoty 0, tuto hodnotu je třeba vynechat. Výše uvedený postup se nazývá smíšený lineární kongruentní generátor a pro posloupnost generovaných čísel platí, že je konečná s periodou $P \leq m$ (Knuth, 1997). V případě $c = 0$ se jedná o multiplikativní kongruentní generátor (Downham and Roberts, 1967). Pokud by platilo, že:

$$x_{n+1} = (x_n + x_{n-1}) \cdot (\text{mod } m) \quad (4.34)$$

jedná se o zpožděný Fibonacciho generátor (Marsaglia, et al., 1990).

V reálných aplikacích se ovšem často vyskytují proměnné, které rovnoměrné rozdělení pravděpodobností nesledují. V těchto případech je tedy nutné převést náhodná čísla rovnoměrného rozdělení na hodnoty náhodné proměnné jiného rozdělení pomocí nějaké metody. Hojně využívanou metodou je inverzní transformace [viz např. Kelton and Law (2000) či Glasserman (2003)]. Metodu inverzní transformace je možné použít v případech, kdy $f(x) > 0$ pro $a < x < b$ a $f(x) = 0$ pro x ostatní. Pak $F(x)$ musí být v intervalu (a, b) rostoucí a jednoznačně každému $x = (a, b)$ přiřazuje hodnotu $r = (0, 1)$. V případě, že je možné najít inverzní funkci k funkci F , lze psát, že $x = F^{-1}(r)$. Pokud bychom chtěli generovat například náhodná čísla exponenciálního rozdělení proměnné, která popisuje vstup požadavků na obsluhu do obslužného systému (tj. čas mezi vstupem dvou po sobě jdoucích požadavků na obsluhu do systému t), měl by vztah pro výpočet této proměnné metodou inverzní transformace následující podobu:

$$t = -\frac{1}{\lambda} \ln(1 - r) \quad (4.35)$$

kde λ je intenzita vstupu požadavků vyjadřující průměrný počet požadavků na obsluhu, který vstoupí do systému obsluhy za jednotku času. V případě, že bychom chtěli metodou inverzní transformace generovat například náhodná čísla trojúhelníkového rozdělení proměnné, která popisuje trvání obsluhy t' , bylo by třeba využít následujících dvou vztahů:

$$t' = a + \sqrt{r(b-a)(m-a)} \rightarrow 0 \leq r \leq \frac{m-a}{b-a} \quad (4.36)$$

$$t' = b - \sqrt{(1-r)(b-a)(b-m)} \rightarrow \frac{m-a}{b-a} \leq r \leq 1 \quad (4.37)$$

kde a představuje optimistický odhad trvání obsluhy (tj. minimální čas trvání obsluhy), b představuje pesimistický odhad trvání obsluhy (tj. maximální čas trvání obsluhy) a m představuje odhad nejpravděpodobnějšího trvání obsluhy.

Další metodou, kterou lze využít ke generování hodnot náhodné proměnné sledující jiné než rovnoměrného rozdělení, je metoda vylučovací [viz např. Dlouhý, et al. (2007)]. Vylučovací metoda pracuje s předpokladem, že hustota pravděpodobnosti $f(x)$ náhodné

proměnné x je v intervalu (a, b) shora ohraničená hodnotou c a vně tohoto intervalu je $f(x)$ rovna nule. Generujeme-li body $[x; y]$ s rovnoměrným rozdělením v obdélníku určeném body a, b a c , mohou nastat dvě situace:

1. Pokud vygenerovaný bod $[x; y]$ leží v oblasti ohraničené hustotou pravděpodobnosti $f(x)$ a platí nerovnost $y \leq f(x)$, potom je hodnota x vygenerované číslo z rozdělení s danou hustotou pravděpodobnosti $f(x)$.
2. Pokud platí $y > f(x)$, bod $[x; y]$ neleží v oblasti ohraničené hustotou pravděpodobnosti $f(x)$ a hodnota x není z příslušného rozdělení.

Generování náhodných čísel vylučovací metodou lze shrnout do následující posloupnosti kroků:

1. Generuj dvojici náhodných čísel rovnoměrného rozdělení r_1 a r_2 .
2. Spočti $x = a + (b - a) \cdot r_1$ a $y = c \cdot r_2$ [hodnoty x a y jsou rovnoměrně rozdělené v intervalech $(a; b)$ a $(0; c)$].
3. Jestliže $y > f(x)$, vrať se zpět ke kroku 1, v opačném případě je x vygenerované číslo z rozdělení s danou hustotou pravděpodobnosti $f(x)$.

Častým případem je potřeba generovat náhodné proměnné nějakého empirického rozdělení získaného analýzou chování náhodné proměnné za uplynulé období. Údaje o chování této proměnné je účelné uspořádat do podoby následující tabulky:

Tabulka 4.2 Empirické rozdělení četností a pravděpodobností

Interval	Četnost	Pravděpodobnost	Kumulované hodnoty pravděpodobnosti
$0 - x_1$	n_1	$p_1 = n_1 / \sum n_i$	p_1
$x_1 - x_2$	n_2	$p_2 = n_2 / \sum n_i$	$p_1 + p_2$
$x_2 - x_3$	n_3	$p_3 = n_3 / \sum n_i$	$p_1 + p_2 + p_3$
...
$x_{k-1} - x_k$	n_k	$p_k = n_k / \sum n_i$	1
Celkem	$\sum n_i$	1	

Hledání prvků empirického rozdělení pro generované r_i spočívá v nalezení takového x_i , pro které je splněna nerovnost:

$$\sum_{k=1}^{x_{i-1}} p_k \leq r_i \leq \sum_{k=1}^{x_i} p_k \quad (4.38)$$

Při stanovení příslušných hodnot se používají většinou středy jednotlivých intervalů.

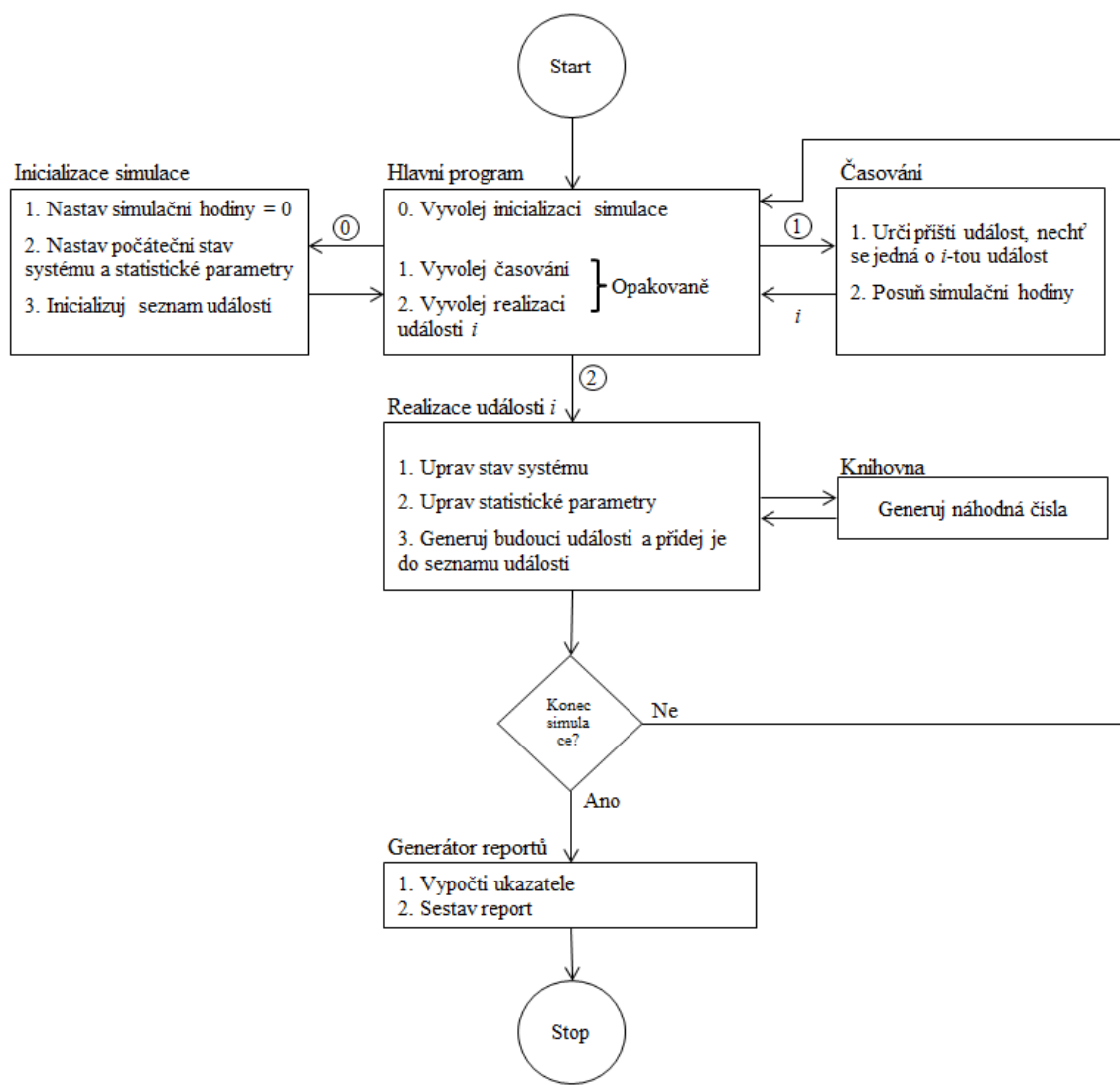
4.3.6 Druhy simulace

4.3.6.1 Discrete event simulace

Discrete event simulace (DES) modeluje dynamiku systému způsobem, kdy ke změně stavových proměnných, které systém popisují, dochází v omezeném počtu časových okamžiků (Kelton and Law, 2000). V těchto okamžicích dochází k realizaci událostí (events), přičemž za událost je považována akce, která může změnit stav systému. Ačkoli DES lze realizovat i pomocí ručních výpočtů, množství vstupních dat, která do simulace vstupují, vyžaduje obvykle zpracování pomocí počítače. DES umožňuje zrychlení času (next-event time advance), kdy simulace přeskakuje časové intervaly, ve kterých se nevyskytuje žádná událost. Ačkoli byla DES aplikována v mnoha různých reálných případech a v mnoha různých průmyslových odvětvích, modely vytvořené touto technikou sdílejí řadu společných komponent a jejich logické uspořádání je stejné bez ohledu na použitý programovací jazyk (např. C, Visual basic (VB), Visual logic (VL) aj.). Základní komponenty DES tvoří:

- Stav systému – stavové proměnné nutné k popisu systému v určitém časovém okamžiku,
- Simulační hodiny – proměnná, která vrací aktuální čas při běhu simulace,
- Seznam událostí – seznam, který obsahuje časy, v nichž dojde k události,
- Statistické parametry – proměnné, které obsahují informace o výkonu systému,
- Inicializace simulace – subprogram, který inicializuje simulaci v čase 0,
- Časování – subprogram, který určuje, která další událost ze seznamu událostí přijde nejdříve na řadu, a posouvá simulační hodiny na čas jejího zahájení,
- Realizace události – subprogram, který mění stav systému při realizaci události,
- Knihovna – subprogramy, které zajišťují generování náhodných čísel pravděpodobnostních rozdělení v případě, kdy je simulace stochastická,
- Generátor reportů – subprogram, který na základě statistických parametrů vytváří report obsahující ukazatele k posouzení výkonu systému na konci simulace,
- Hlavní program – subprogram, který vyvolává časování k určení, která další událost ze seznamu událostí přijde na řadu, a poté předává kontrolu realizaci události, která změní stav systému. Na konci simulace také vyvolává činnost generátoru reportů.

Logické vztahy mezi základními komponentami DES znázorňuje Obr. 4.3.



Obr. 4.3 Uspořádání základních komponent v DES

Modely DES se tradičně používají na taktické a operativní úrovni jako prostředek podpory rozhodování. Velmi užitečnou vlastností těchto modelů je možnost grafického zobrazení modelovaného systému či dokonce jeho převedení do interaktivní podoby. Vzhledem k tomu, že DES pracuje velmi často s náhodnými veličinami, je většinou nezbytné pracovat s rozsáhlou základnou vstupních dat, jejichž sběr a zpracování mohou být časově náročné. Stejně tak výstupy DES je třeba získávat opakováním běhu simulace a jejich statistickým zpracováním, neboť jde rovněž o náhodné veličiny.

Některé aplikace DES v modelování reálných systémů z různých oborů jsou uvedeny v následující tabulce.

Tabulka 4.3 Aplikace DES v modelování reálných systémů

Publikace	Stručný popis aplikace	Odvětví
Mendes, et al. (2005)	Konfigurace výrobní linky a její vybalancování pro různé intenzity materiálových toků ve výrobě fotoaparátů.	Výroba elektroniky
Weston Jr, et al. (1999)	Analýza procesů a plánování výroby v mikropivovaru.	Potravinářství
Roser, et al. (2006)	Lokalizace a stanovení velikosti zásobníků ve výrobním procesu s cílem maximalizace využití úzkého místa.	Automobilový průmysl
Blocher, et al. (1999)	Optimalizace stavu zásob a průběžné doby výroby pro různé úrovně služeb zákazníkům.	Výroba léčiv
Bozarth and Berry (1997)	Posouzení vlivu strategie řízení výroby na schopnost podniku uspokojit požadavky zákazníků.	Výroba nábytku
Cachon and Fisher (1997)	Návrh VMI konceptu ve 2stupňovém dodavatelském systému dodavatel – výrobce.	Potravinářství
Fleisch and Tellkamp (2005)	Posouzení vlivu kvality informačních toků na výši zásob v 3stupňovém dodavatelském systému dodavatel – maloobchodník – zákazník.	Maloobchodní prodej potravin
Butler, et al. (2001)	Plánování a řízení projektu na základě různých kritérií.	Petrochemie
Spedding and Sun (1999)	Aplikace activity based costing při alokaci nákladů na výrobní lince.	Elektrotechnika
Ayag (2005)	Vývoj nového produktu.	Strojírenství

4.3.6.2 System dynamics

System dynamics (SD) je metoda modelování systémů, jejíž základy položil ve svých pracích Jay W. Forrester (viz Forrester (1958) a Forrester (1960)). Základem této metody je modelování entit v systému pomocí spojitých toků (flows) a zásob (stocks). Na rozdíl od DES dochází ke změně stavu systému v čase kontinuálně. Konceptuální model SD má obvykle podobu diagramu kauzálních smyček (causal loop diagram; influence diagram), což je nástroj pro grafické znázornění a analýzu vzájemných vztahů proměnných (Chahal and Eldabi, 2010). Vizualizace vzájemných vztahů proměnných usnadňuje pochopení funkce komplexních systémů a pomáhá tyto vztahy kvantifikovat. Diagram kauzálních smyček se skládá z množiny uzlů reprezentujících proměnné a množiny šipek znázorňujících vztahy mezi nimi (viz Obr. 4.4).

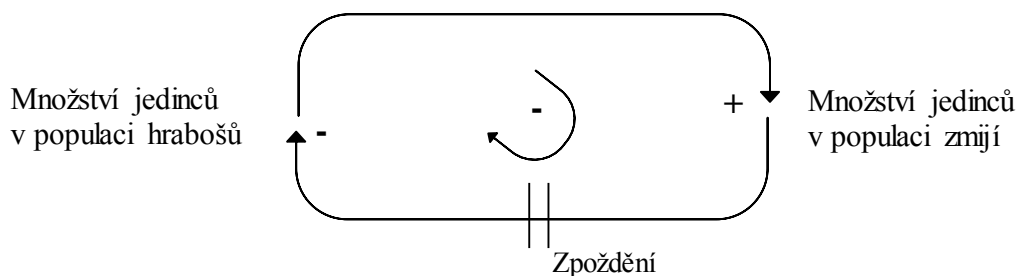


Obr. 4.4 Diagram kauzálních smyček s pozitivní závislostí proměnných

Proměnná u začátku šipky je kauzální příčinnou změny proměnné na konci šipky. Znaménko plus vyjadřuje pozitivní závislost, kde vzrůst respektive pokles první proměnné vyvolá růst respektive pokles druhé proměnné. Znaménko mínus vyjadřuje negativní závislost, kde růst respektive pokles první proměnné vyvolá pokles respektive růst druhé proměnné. Každá šipka v diagramu vyjadřuje hypotézu o závislosti dvou proměnných. Řetěz kauzálních vztahů vytváří kauzální smyčku v situaci, kdy změna jedné proměnné ovlivní jednu nebo více dalších proměnných, které pak zpětně ovlivní hodnotu původní proměnné. Smyčky se zpětnou vazbou se rozlišují na (Brailsford and Hilton, 2001):

- posilující smyčky (reinforcing loops), které se označují znaménkem plus,
- rovnovážné smyčky (balancing loops), které se označují znaménkem mínus.

Posilující smyčky jsou spojeny s exponenciálním růstem hodnot proměnných, zatímco rovnovážné smyčky jsou spojeny s dosažením rovnovážných hodnot. V modelech, které implementují zpoždění ve vývoji některé proměnné, dochází k oscilaci (viz Obr. 4.5).



Obr. 4.5 Diagram kauzálních smyček s rovnovážnou smyčkou a zpožděním

Diagram na obrázku modeluje vývoj populací predátora a kořisti (viz např. Gordon (1978); Braun and Golubitsky (1983)). Nárůst počtu kořisti (hrabošů) umožní nárůst počtu

predátorů (zmijí), kteří pak zpětně snižují počty kořisti. Příčinou zpoždění je čas, který uplyne, než mládřata predátorů dospějí, a také to, že nedostatek kořisti nezpůsobí okamžité uhynutí predátora. Kdyby v diagramu nebylo zpoždění, dosáhl by počet predátorů i kořisti rovnovážné hodnoty a více by se neměnil, ale díky zpoždění způsobí nárůst počtu kořisti větší nárůst počtu predátorů, než je optimální. Ti pak příliš sníží počet kořisti a po určitém čase začnou hynout hlady. Vše se opakuje a výsledkem je oscilující vzorec chování. Vztahy mezi proměnnými v diagramu jsou popsány pomocí diferenciálních rovnic, k jejichž řešení se obvykle využívá numerická integrace.

Zatímco DES se primárně využívá na operativní a taktické úrovni, SD k podpoře rozhodnutí na strategické úrovni či k hlubšímu pochopení fungování systému. SD také většinou pracuje s deterministickými daty.

Některé aplikace SD v modelování reálných systémů z různých oborů jsou uvedeny v následující tabulce.

Tabulka 4.4 Aplikace SD v modelování reálných systémů

Publikace	Stručný popis aplikace	Odvětví
Owens and Levary (2002)	Návrh uspořádání výrobní linky ve výrobě sušených potravin s cílem zvýšení průtoku.	Potravinářský průmysl
Lyneis (2000)	Předpověď poptávky průmyslového odvětví a porovnání její úspěšnosti s tradičními předpovědními metodami.	Výroba letadel
Ashayeri, et al. (1998)	Reengineering podnikových procesů.	Neuvedeno
Spengler and Schröter (2003)	Návrh strategie výroby, distribuce a recyklace náhradních dílů v CLSC.	Výroba elektroniky
Wenzler (2005)	Návrh a ověření účinnosti strategie pro řízení aktiv podniku.	Energetický průmysl
Anderson Jr, et al. (2000)	Plánování a řízení poptávky se sezónností v dodavatelském systému pro výrobu a distribuci strojních zařízení.	Strojírenský průmysl
Wolstenholme (2003)	Návrh metody pro oceňování nových technologií a její porovnání s tradičními přístupy k této problematice.	Výroba léčiv
Rodrigues and Williams (1998)	Stanovení dopadu chování zákazníků na výstupy projektu.	Vývoj softwarových aplikací
Schwaninger, et al. (2006)	Srovnání role manažera a konzultanta při ovlivňování podnikových procesů a dopad jejich rozhodnutí na výkonnost podniku.	Výroba léčiv

4.3.6.3 Ostatní simulační metody

Agent based simulace je založena na vzájemné interakci agentů při změně vnějších podmínek (Swaminathan, et al., 1998). Agenti disponují vnitřními stavy, které je možné popsat atributy či proměnnými. Tyto proměnné mohou být jak spojité, tak diskrétní. Při vzájemné interakci agentů v modelovaném systému dochází ke změnám vnitřních stavů popsaných pomocí atributů a proměnných. Předmětem zájmu je schopnost adaptace agentů na vnější změny v průběhu času (Macal and North, 2010).

Inteligentní simulace kombinuje simulační metody s umělou inteligencí. Využívá se zejména v optimalizačních úlohách v rámci plánování či designu dodavatelských systémů.

Simulace Monte Carlo představuje velmi jednoduchou metodu, která je založena na generování náhodných čísel rovnoměrného rozdělení (Glasserman, 2003). Využívá se především v úlohách z oblasti financování a risk managementu (viz například Fishman (2005), Fotr, et al. (2013)).

V případě kombinace simulací DES a Monte Carlo v prostředí Microsoft Excel hovoříme o spreadsheet simulaci. Metodu lze opět aplikovat pouze na nepříliš složité úlohy s malým množstvím vstupních dat (Evans and Olson, 1998). Tato technika se využívá například v oblastech financování, plánování poptávky, plánování výroby nebo projektového managementu.

Simulační hry představují skupinu simulací používanou především pro výukové účely. Jedná se o interaktivní aplikace zaměřené zpravidla na specifickou oblast podniku, jako je logistika, marketing či finanční řízení. Příkladem jsou například hry Markstrat² či Marketplace.³

Virtuální simulace nabízí uživatelům možnost zahrnout do simulačního modelu 2D či 3D grafiku. Využívá se především k lepšímu pochopení logiky modelovaného systému či k názorné prezentaci dopadu změn na modelovaný systém. 2D či 3D grafika je v současné době k dispozici v mnoha komerčních softwarových produktech, které využívají DES či SD. Podrobněji o těchto produktech pojednám v následující části práce.

Hybridní simulace kombinuje několik druhů simulace dohromady. Nejčastější je kombinace DES a SD. Využívá se v úlohách, kdy je třeba kombinovat rozhodování na taktické a operativní úrovni s rozhodováním na úrovni strategické (Rabelo, et al., 2005).

Skupina simulačních technik vyvinutá speciálně pro modelování dopravních systémů se nazývá dopravní simulace. Využívá se například pro modelování provozu na komunikacích, řízení křižovatek pomocí semaforů apod.

Následující tabulka uvádí některé aplikace ostatních simulačních metod v modelování reálných systémů z různých oborů.

² <http://web.stratxsimulations.com/simulation/strategic-marketing-simulation/>; citováno 2. 12. 2017.

³ <http://www.marketplace-simulation.com/>; citováno 2. 12. 2017.

Tabulka 4.5 Aplikace ostatních simulačních metod v modelování reálných systémů

Publikace	Typ simulace	Stručný popis aplikace	Odvětví
Zha and Lim (2000)	Inteligentní	Plánování výroby čoček.	Výroba optických zařízení
Orady, et al. (1997)	Virtuální	Nástroj pro 3D zobrazení výrobních linek s robotizací a jeho využití k vizualizaci navrženého uspořádání výroby před samotnou investicí do výstavby.	Automobilový průmysl
Lee, et al. (2003)	Hybridní	Řízení dopravy v městech.	Doprava
Lin and Pai (2000)	Agent based	Design podnikových procesů.	Výroba dílů
Zenios, et al. (1998)	Monte Carlo	Řízení portfolia cenných papírů za nejistoty.	Obchod
Hoogeweegen, et al. (2006)	Simulační hry	Výuková hra pro studenty, jejímž cílem je tvorba strategie podniku.	Bankovníctví
Boel and Mihaylova (2006)	Dopravní	Online plánování tras s ohledem na aktuální dopravní situaci.	Doprava

4.3.7 Význam simulace v modelování dodavatelských systémů

4.3.7.1 Význam simulačních metod v modelování dodavatelských systémů – srovnání s jinými modelovými přístupy

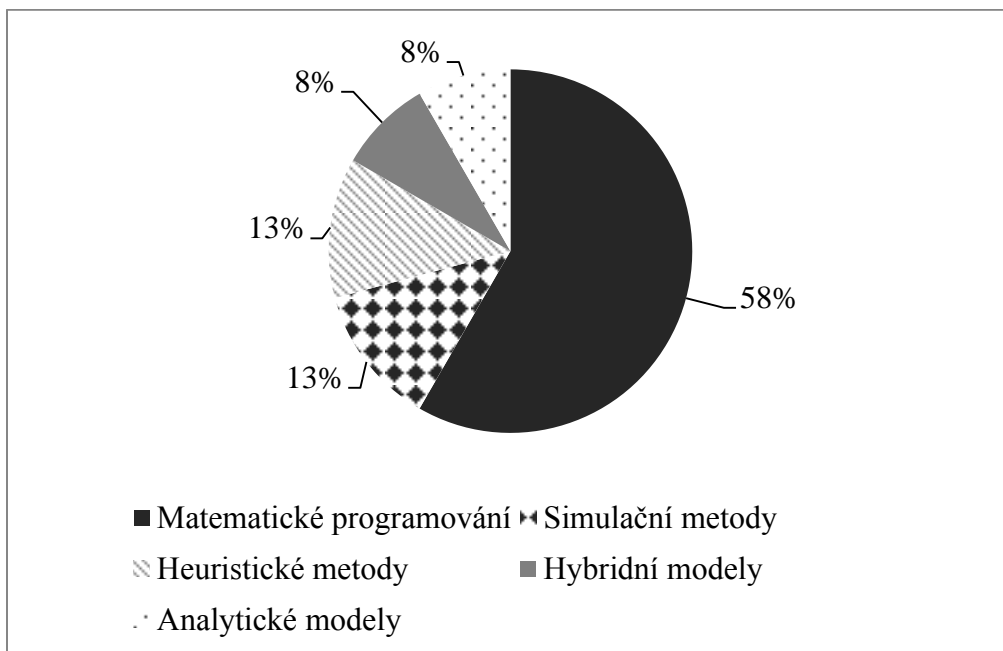
Pro stanovení významu simulačních metod v modelování dodavatelských systémů v porovnání s jinými modelovými přístupy byly využity 4 literární rešerše různých autorů, které jsou popsány v Tabulka 4.6. První rešerše pokrývá období 1981–1997, vzorek relevantních prací je poměrně malý a není uveden způsob jeho výběru. Druhá rešerše pokrývá období 1996–2010 a při výběru relevantních prací využívá obsahovou analýzu (viz Mayring and Gläser-Zikuda (2008); Krippendorff (2012)). Rešerše Brandenburg, et al. (2014) pokrývá období 1994–2012 a při výběru relevantních prací pracuje s obsahovou analýzou a vyhledáváním pomocí klíčových slov v databázích Emerald, Elsevier, Wiley, Springer, Ebsco, Scopus a Metapress. Vzorek relevantních prací v této rešerši je dále zúžen požadavkem na citování prací v alespoň 5 ze 6 časopisů, které jsou považovány za

nejkvalitnější pro publikování v oblasti modelování dodavatelských systémů. Jedná se o časopisy Decision Support Systems, European Journal of Operational Research, International Journal of Production Economics, International Journal of Production Research, Journal of Cleaner Production a Transportation Research Part E (Hassini, et al. (2012); Tang and Zhou (2012)). Čtvrtá rešerše pokrývá období 1997–2006 a při výběru relevantních prací pracuje s obsahovou analýzou a vyhledáváním pomocí klíčových slov v databázi Scopus.

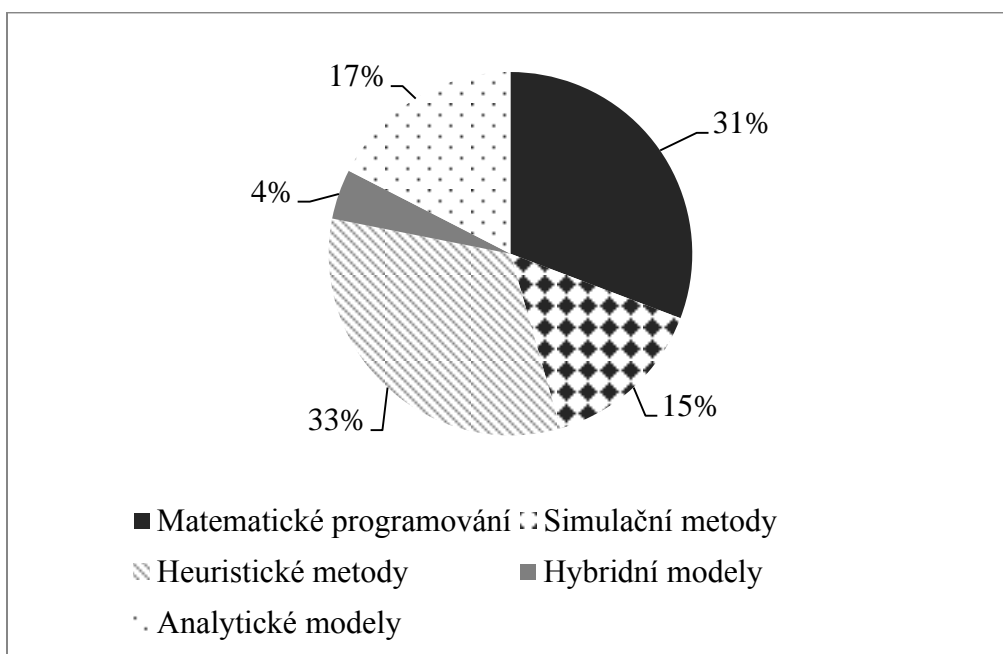
Tabulka 4.6 Literární rešerše využité při stanovení významu simulačních metod v modelování dodavatelských systémů

Rešerše	Pokrývá období	Počet prací	Vyhledávání prací na základě klíčových slov – obsahová analýza
Beamon (1998)	1981–1997	24	neuvedeno – neuvedeno
Ilgın and Gupta (2010)	1996–2010	540	ne – ano
Brandenburg, et al. (2014)	1994–2012	134	ano – ano
Jahangirian, et al. (2010)	1997–2006	28	ano – ano

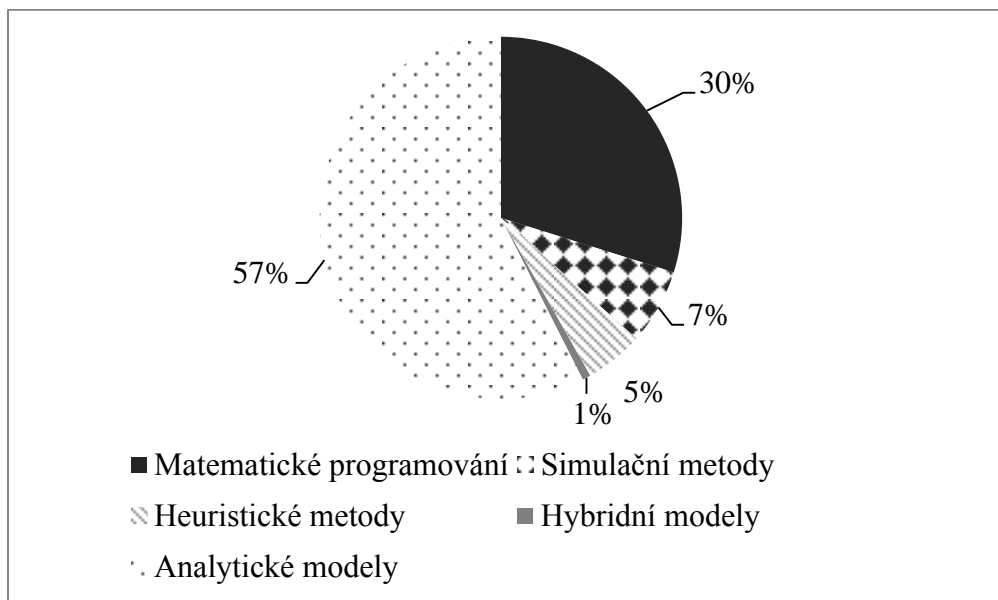
Podíl využití modelových přístupů na celkovém počtu relevantních prací zahrnutých do literárních rešerší zachycují Obr. 4.6, Obr. 4.7 a Obr. 4.8. Zastoupení článků v literárních rešerších, které pracují s nějakou simulační metodou, je 13 % a v případě rešerší, které nezohledňují minimální počty citací těchto článků v nejkvalitnějších vědeckých časopisech, je to 15 %. V případě zohlednění počtu citací v nejkvalitnějších vědeckých časopisech zastoupení dále klesá na 7 %. Příčiny relativně nízkého podílu publikací, jejichž jádro tvoří simulační metody, je dáno především samotnou povahou dynamické simulace. V případě DES, která v publikacích převládá (viz následující kapitola), je velmi obtížné zajistit reprodukovatelnost závěrů výzkumu, a to především proto, že simulační model je tvořen programovým kódem, který je obvykle velmi rozsáhlý a silně závislý na použitém softwarovém produktu či na přístupu uživatele, který systém modeluje. Typické publikace o modelování dodavatelských systémů, ve kterých je využita DES, jsou proto zaměřeny na formulaci konceptuálního modelu a aplikaci simulační metody či softwaru jen konstatují. Pouze minimum publikací jde do většího detailu a věnuje se například popisu modulární struktury simulace, která napomáhá výraznému zrychlení procesu stavby modelu. Tyto publikace se objevují především ve specializovaných časopisech či na úzce zaměřených konferencích.



Obr. 4.6 Podíl využití modelových přístupů na celkovém počtu relevantních prací zahrnutých do literární rešerše – literární rešerše Beamon (1998)

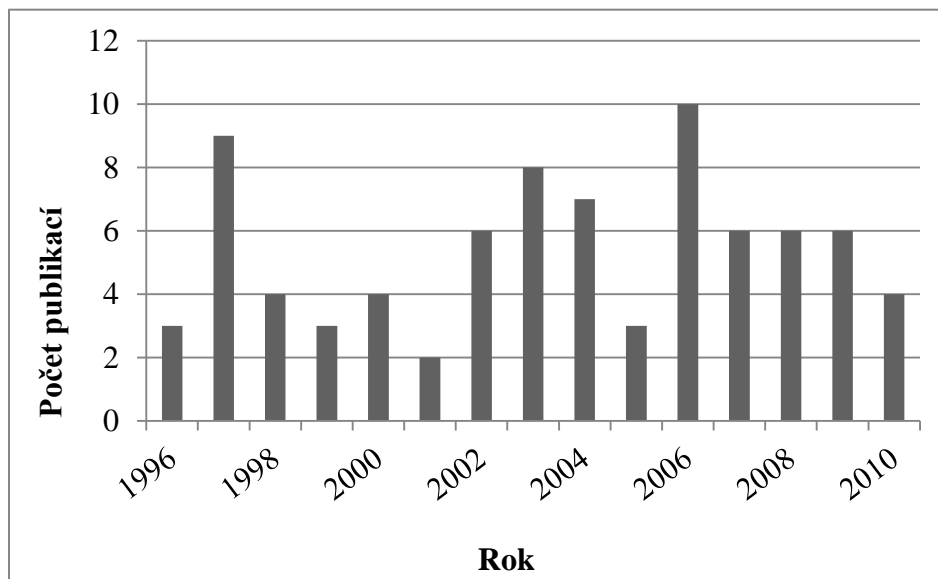


Obr. 4.7 Podíl využití modelových přístupů na celkovém počtu relevantních prací zahrnutých do literární rešerše – literární rešerše Ilgin and Gupta (2010)

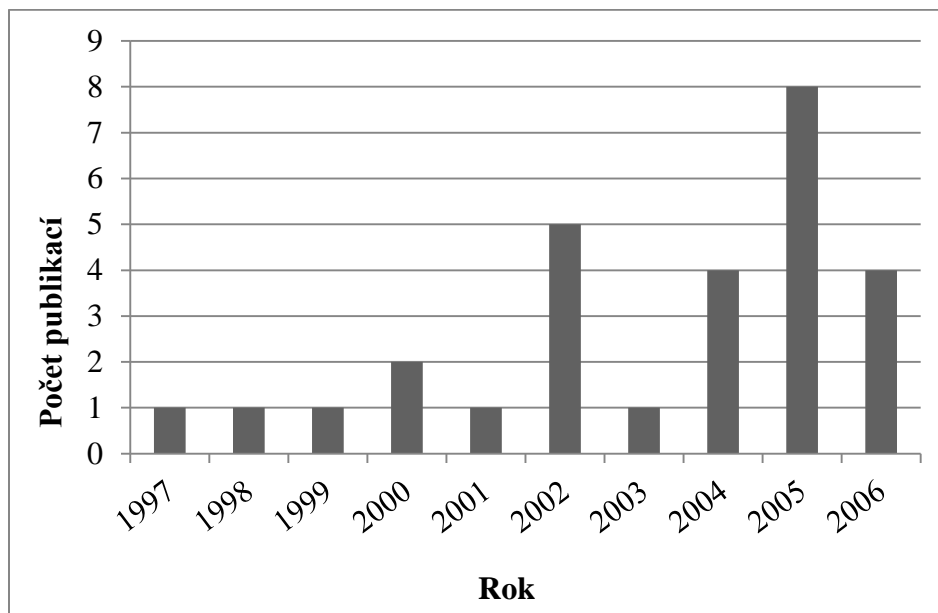


Obr. 4.8 Podíl využití modelových přístupů na celkovém počtu relevantních prací zahrnutých do literární rešerše – literární rešerše Brandenburg, et al. (2014)

Výše uvedené skutečnosti se promítají také do vývoje počtu publikací o modelování dodavatelských systémů, které pracují s nějakou simulační metodou (viz Obr. 4.9 a Obr. 4.10). Zatímco do roku 2006 je možné pozorovat pozvolný nárůst počtu publikací, po roce 2006 dochází k zastavení tohoto trendu.



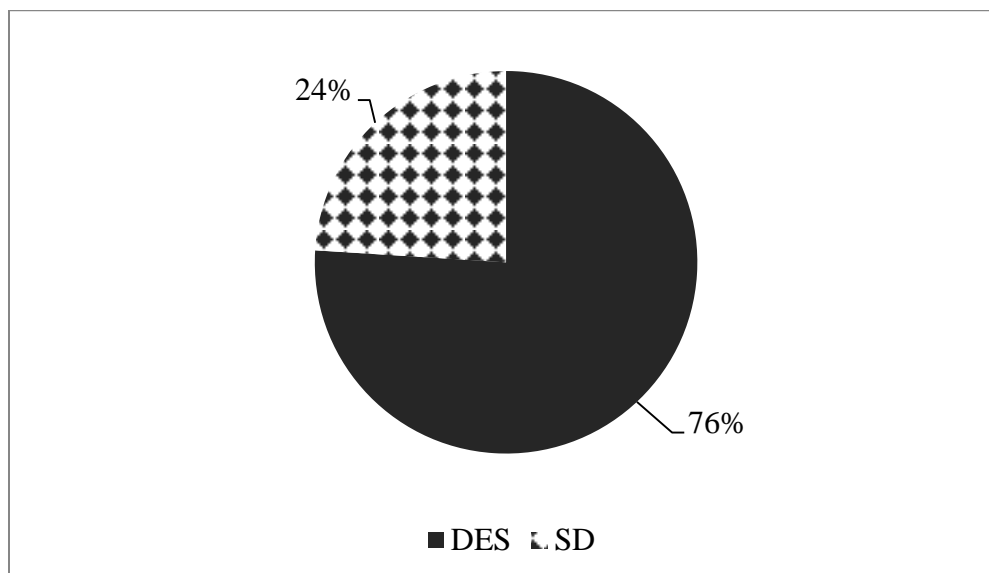
Obr. 4.9 Počet publikací o modelování dodavatelských systémů se simulační metodou – literární rešerše Ilgin and Gupta (2010)



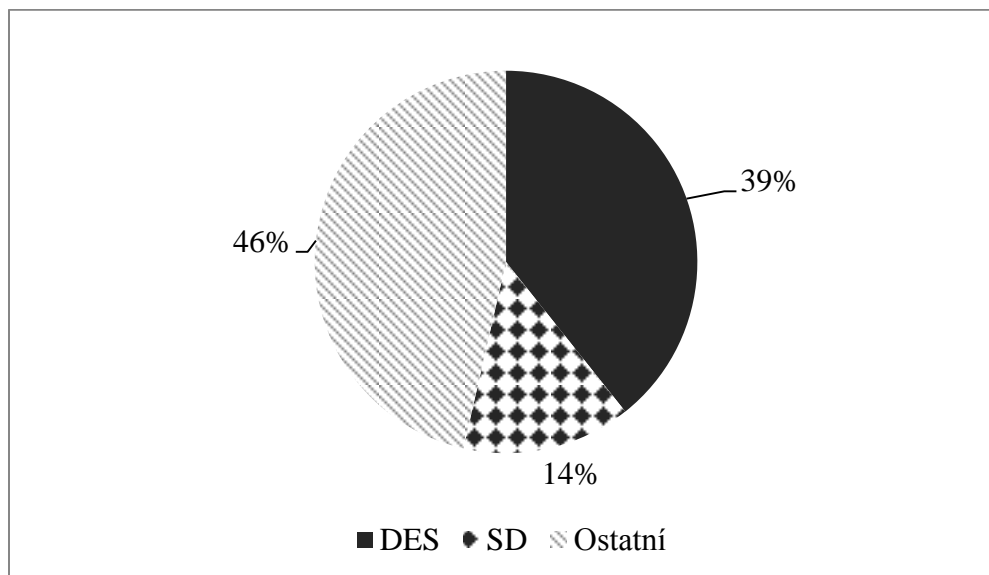
Obr. 4.10 Počet publikací o modelování dodavatelských systémů se simulační metodou – literární rešerše Jahangirian, et al. (2010)

4.3.7.2 Srovnání významu jednotlivých simulačních metod

Podíl využití druhů simulace na celkovém počtu relevantních prací zahrnutých do literární rešerše je zachycen na Obr. 4.11 a Obr. 4.12.



Obr. 4.11 Podíl využití druhů simulace na celkovém počtu relevantních prací zahrnutých do literární rešerše – literární rešerše Ilgin and Gupta (2010)



Obr. 4.12 Podíl využití druhů simulace na celkovém počtu relevantních prací zahrnutých do literární rešerše – literární rešerše Jahangirian, et al. (2010)

Je zřejmé, že výrazně převládá využití DES nad SD, velmi časté jsou také aplikace, ve kterých je využit nějaký speciální typ simulace.

5 SIMULAČNÍ SOFTWARE

Na současném trhu je k dispozici celá řada simulačních balíčků, které mohou být využity jako nástroje pro podporu rozhodování. Tyto produkty se liší zejména v následujících bodech (Abu-Taieh and El Sheikh, 2007):

- způsob modelování systémů,
- využití tabulkových procesorů pro tvorbu reportů,
- programovací jazyk,
- způsob nakládání s časem simulace a
- možnost 2D či 3D animace.

Hlupic, et al. (1999) navrhli metodiku výběru simulačního softwaru založenou na 13 skupinách kritérií (viz Tabulka 5.1). Kritéria jsou stanovena na základě dotazování, přičemž odpovědi na otázky jsou přesně definovány.

Tabulka 5.1 Kritéria metodiky výběru simulačního softwaru

Kritéria	Popis	Počet otázek
Obecný popis	Typ simulace; způsob modelování systému; oblast využití; obtížnost práce se softwarem aj.	26
Vizualizace simulace	2D a 3D grafika; animace; virtuální realita; možnost editace a importu ikon; dostupnost a rozsah knihovny ikon aj.	37
Programový kód	Programovací jazyk; generátor programového kódu; dostupnost proměnných; způsob nakládání s časem v simulaci; načítání vstupních dat; psaní komentářů aj.	24
Efektivita modelu	Robustnost; úroveň detailu; rychlost běhu simulace; možnost interaktivního ovládání experimentů; možnost definice toku entit pomocí myši aj.	26
Asistence při modelování	Funkce Prompt; modularita; možnost vrátit se o krok zpět; modelování pomocí myši; on-line nápověda, její kvalita a dostupnost aj.	21
Ladění modelu	Interaktivní zobrazení chyb v programovém kódu; krokování běhu simulace; možnost zobrazení proměnných a atributů; okamžité uživatelské akce aj.	27
Kompatibilita softwaru	Integrace s tabulkovými procesory, textovými editory, databázemi a informačními systémy aj.	8
Vstupy a výstupy simulace	Pull down menu; způsob načítání vstupních dat do modelu; statické a dynamické grafické výstupy; histogramy, koláčové grafy; dynamické grafy aj.	25
Experimentování s modelem	Warm-up perioda; úprava rychlosti běhu simulace; přerušení běhu simulace; posouzení přesnosti výpočtů; dostupnost pomocníka při designu experimentu aj.	11
Statistická kritéria	Teoretická a empirická rozdělení pravděpodobností; generování náhodných čísel pomocí generátoru; intervalový odhad; test dobré shody; statistická analýza výstupů aj.	12
Uživatelská podpora	Velikost komunity uživatelů; demo modely a uživatelská příručka; četnost a trvání školicích kurzů; údržba softwaru a jeho upgrade na vyšší verzi aj.	20
Finanční a technické	Cena a náklady na instalaci softwaru; přenosnost licence; síťová licence a způsob jejího zabezpečení; sleva na	21

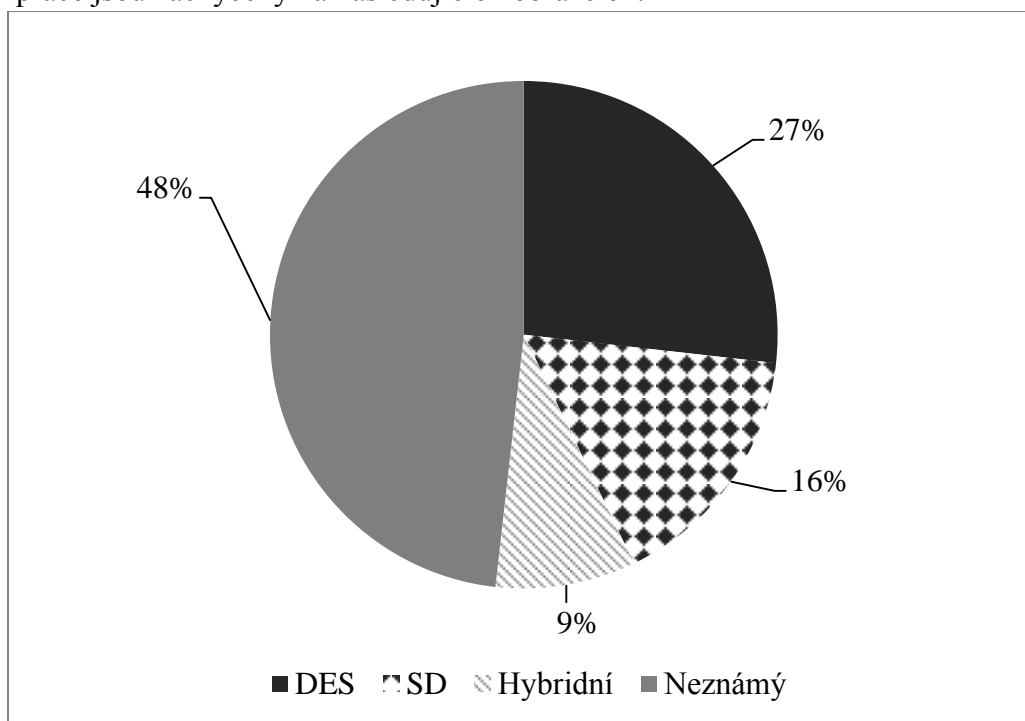
parametry	licenci pro výukové účely; požadavky na hardware a operační systém aj.
-----------	--

Ostatní kritéria	Stáří produktu; pověst dodavatele; reference uživatelů; dostupné zdroje informací o softwaru aj.
------------------	--

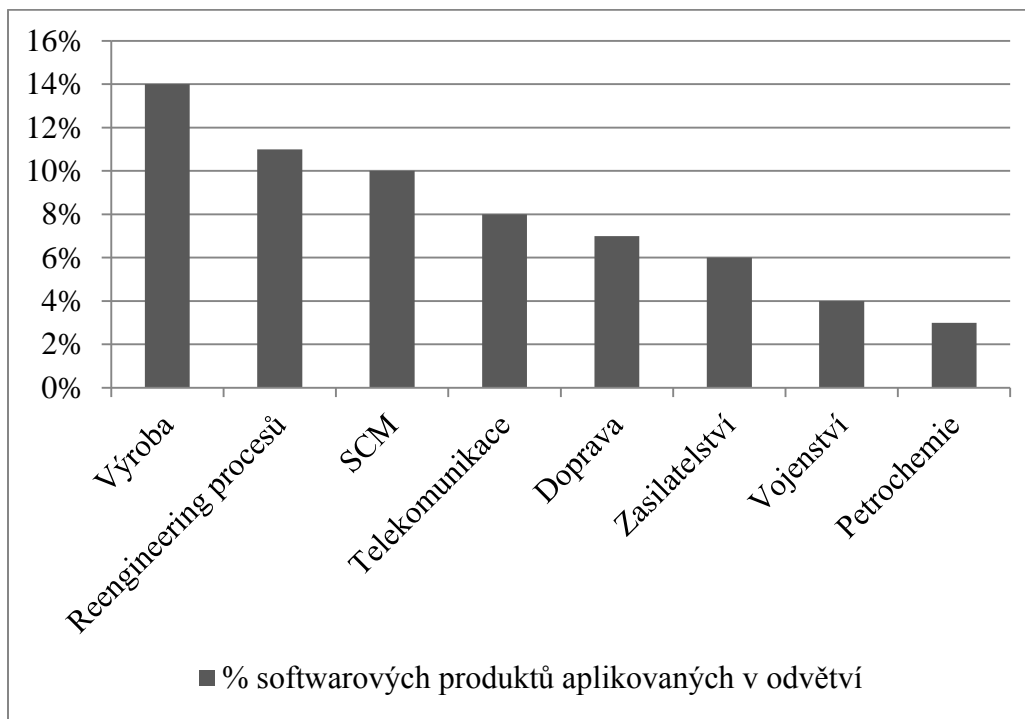
8

Fu (2002) provedl srovnání 6 modulů pro optimalizaci modelů vytvořených v prostředí simulačních balíčků. Ve studii jsou popsány algoritmy používané při optimalizaci a výhody a nevýhody jejich použití.

Abu-Taieh and El Sheikh (2007) rozřídili 56 softwarových produktů do skupin dle různých charakteristik. Třídění bylo provedeno pomocí informací uvedených o produktech na webových stránkách vývojářských společností a analýzy vědecké literatury. Výsledky jejich práce jsou zachyceny na následujících obrázcích:



Obr. 5.1 Rozřídění softwarových produktů dle druhu simulace



Obr. 5.2 Aplikace softwarových produktů v odvětvích popsané ve vědecké literatuře

Cimino, et al. (2010) provedli srovnání 7 simulačních produktů pomocí 12 kritérií. Jednalo se o produkty Anylogic, Arena, AutoMod, Emplant, Promodel, Flexsim a Witness. Kritéria byla vypočtena na základě odpovědí získaných pomocí dotazování 100 odborníků na práci v prostředí dynamické simulace. Nejlépe hodnocenými produkty byly Arena a Witness.

5.1 Software pro discrete event simulaci

Počátek vývoje softwaru pro DES se datuje do 60. let 20. století. V současné době jsou k dispozici simulační produkty čtvrté generace (Wang, et al., 1993). První generace softwaru DES je založena na modelování logiky a akcí pomocí programovacího jazyka Fortran. Druhá generace rozvíjí funkčnost generace předchozí a nabízí možnosti v oblasti generování náhodných čísel a tvorby reportů. Do této skupiny patří například GPSS, See Why a AutoMod. Simulační software třetí generace využívá generátor kódu, který výrazně snižuje čas potřebný na vytvoření modelu. Patří sem například Siman nebo Express. Produkty první, druhé a třetí generace bohužel vyžadují rozsáhlé znalosti z modelování a informatiky a nejsou příliš přístupné manažerům jako nástroj podpory rozhodování na denní bázi. Produkty čtvrté generace byly uvedeny na trh po roce 1980. Jedná se o interaktivní simulace s paletou předdefinovaných elementů, které lze intuitivně spojovat

často bez použití programovacího jazyka pro zachycení logiky systému. Příkladem jsou simulační balíčky Witness, Simul8 a Arena.

5.1.1 Arena

Jedním z nejrozšířenějších simulačních nástrojů na současném trhu je Arena.⁴ Jde o kompletní a flexibilní modelovací prostředí s intuitivním uživatelským rozhraním. Umožňuje navrhovat počítačové modely, které přesně vystihují existující nebo navrhované procesy. Arena integruje všechny potřebné funkce (tj. vstupní datovou analýzu, vizualizaci modelu, validaci modelu, výstupní analýzu atd.) do jednoho simulačního modelovacího prostředí. Celkovým výsledkem po zpracování modelu je animace, která naprosto reálně zobrazuje průběh, vazby a zákonitosti celého modelu. Animaci je možné krokovat a zastavovat u jednotlivých klíčových momentů. Současně je v animaci viditelné průměrné vytížení všech souvisejících zdrojů, probíhající čas a řada dalších charakteristik. Celý systém je kompatibilní s Microsoft Office a je podporován programovacím jazykem VB, což umožňuje značnou pružnost při konkrétních řešeních.

Základní elementy dostupné v prostředí Arena se nazývají moduly. Jedná se o:

1. Modul Create

Jedná se o modul, který umožňuje vstup entit do modelu. Umisťuje se na začátek toku entit.

2. Modul Dispose

Opak modulu Create. Jeho funkcí je odstranění entit z modelu na konci jejich toku.

3. Modul Process

Slouží k simulování aktivit, které zabírají určité množství času. K realizaci těchto aktivit je obvykle nutné využít nějaké zdroje.

4. Modul Decide

Je využíván k větvení toku entit na základě logických podmínek.

V následující tabulce jsou uvedeny odkazy na literaturu, ve které se objevují některé aplikace simulačního nástroje Arena v modelování dodavatelských systémů.

⁴ <https://www.arenasimulation.com/>; citováno 5. 12. 2017.

Tabulka 5.2 Aplikace simulačního nástroje Arena v modelování dodavatelských systémů

Publikace	Stručný popis aplikace	Odvětví
Pan (2008)	Stanovení výkonu rafinerie v podmínkách náhodné poptávky po ropných produktech; návrh optimální strategie zásobování.	Petrochemie
Cafaro, et al. (2010)	Operativní plánování distribuce ropných produktů v potrubní síti.	Petrochemie
Xu, et al. (2011)	Simulace dopadu změn a analýza rizika ve výzkumu a vývoji nových komponent v leteckém průmyslu.	Strojírenství
Ammeri, et al. (2013)	Stanovení optimální velikosti výrobní dávky v make to order výrobním systému.	Neuvedeno
Muresan, et al. (2014)	Hodnocení výkonnosti dodavatelského systému výroby a distribuce vodíku pocházejícího z biomasy v podmínkách nejistoty.	Chemie
Hailu, et al. (2015)	Návrh a simulace distribučního systému výrobků z kůže s cílem zkrácení termínu vyřízení objednávky a minimalizace distribučních nákladů.	Kožený průmysl
Azougagh, et al. (2016)	Obecný model dodavatelského systému vytvořený na principu Petriho sítí; stanovení výkonnosti dodavatelského systému.	Neuvedeno

5.1.2 Simul8

Simul8⁵ je produktem firmy Simul8 a je určen pro modelování podnikových procesů. Model je v něm sestavován z předdefinovaných elementů, které je možné propojit pomocí spojnic, znázorňujících logické vazby. Na vstupu i výstupu těchto elementů je možné zvolit pravidla pro řízení toku entit modelem. Program disponuje několika sadami stylů vizualizace pro různá prostředí (obslužné systémy, výroba). Základní vazby modelu jsou definovány v dialogových oknech elementů. Pokročilejší nastavení je možné pomocí příkazů v prostředí programovacího jazyku VL (jazyk na bázi VB), kde je možné vytvářet podmíněné příkazy a cykly. Ovládání je velmi intuitivní, pro uživatele zvyklého na práci v prostředí Windows je po krátkém zaškolení snadné tvořit jednoduché modely. Simul8 lze propojit s dalšími programy jako MS Excel, MS Access či s databázemi využívajícími SQL pro načítání vstupních dat nebo export výstupů.

⁵ <http://www.simul8.com/>; citováno 5. 12. 2017.

Základní předdefinované elementy dostupné v prostředí Simul8 jsou:

1. Work item

Work item představuje entitu, která vstoupí do modelu, pohybuje se modelem, vyvolává různé akce, spotřebovává zdroje a na konci opouští model. Jedná se například o výrobek či zákazníka.

2. Work entry point

Work entry point představuje objekt, který umožňuje entitám vstup do systému. Používá se například k simulaci akcí typu „vstup zákazníka do prodejny“.

3. Work center

Work center se používá k modelování akcí, kterými procházejí entity. Využívá se často v kombinaci se zdroji (element Source), které je nutné spotřebovat při provedení akce. Akce mohou reprezentovat například výrobní operace či proces obsluhy zákazníka.

4. Storage bin

Storage bin reprezentuje zásobník, ve kterém se kumulují entity. Používá se například k modelování front či skladů.

5. Work exit point

Work exit point je objekt, skrz který prochází entita při opuštění modelu.

6. Source

Source se využívá v elementu Work center jako podmínka pro provedení nějaké akce s entitou. Jedná se například o pracovníka, který provádí určitou výrobní operaci.

7. Route

Route se používá k propojení ostatních elementů a udává sekvenci aktivit, kterými procházejí entity.

V následující tabulce jsou uvedeny odkazy na literaturu, ve které se objevují některé aplikace simulačního nástroje Simul8 v modelování dodavatelských systémů.

Tabulka 5.3 Aplikace simulačního nástroje Simul8 v modelování dodavatelských systémů

Publikace	Stručný popis aplikace	Odvětví
Aguirre, et al. (2008)	Návrh a optimalizace dodavatelského systému při výrobě součástek do motorů osobních a nákladních automobilů.	Automobilový průmysl
Williams, et al. (2010)	Návrh a optimalizace záchranného zdravotnického systému při náhodných požadavcích pacientů, plánování kapacit.	Zdravotnictví
Princ, et al. (2011)	Optimalizace materiálových toků mezi výrobní halou a skladem, nalezení úzkých míst.	Neuvedeno
Bevilacqua, et al. (2012)	Řešení problému přetížení leteckých tras a letišť ve stochastických podmínkách, optimalizace nákladů.	Letecký průmysl
Pepino, et al. (2015)	Optimalizace využití zdravotnického personálu v nemocnici.	Zdravotnictví
Armenzoni, et al. (2016)	Optimalizace procesů spojených s výrobou mléčných výrobků.	Potravinářský průmysl
Fousek, et al. (2017)	Nalezení úzkého místa ve výrobním procesu radiálních ventilátorů.	Strojírenství

5.1.3 Witness

Program Witness⁶ společnosti Lanner Group je simulačním softwarem, který bude využit v praktické části práce, a proto se jeho popisu budu věnovat podrobněji. Postup modelování ve Witnessu je velmi jednoduchý a je možné ho rozdělit do 3 kroků. V prvním kroku je definován vybraný element, který je k dispozici v knihovně předdefinovaných elementů. Definice elementu znamená, že prvek je fyzicky přítomen v modelu a lze pro něj zobrazit požadované vlastnosti na obrazovku (krok 2), respektive editovat jeho funkční detaily s ohledem na logiku simulovaných procesů (krok 3).

Knihovna předdefinovaných elementů ve Witnessu disponuje prvky které lze rozdělit do 3 skupin. Jedná se o:

- fyzické elementy,
- logické elementy,

⁶ <http://www.lanner.com/en/witness.cfm>; 5. 12. 2017.

- grafické elementy.

Mezi fyzické elementy patří:

1. Part

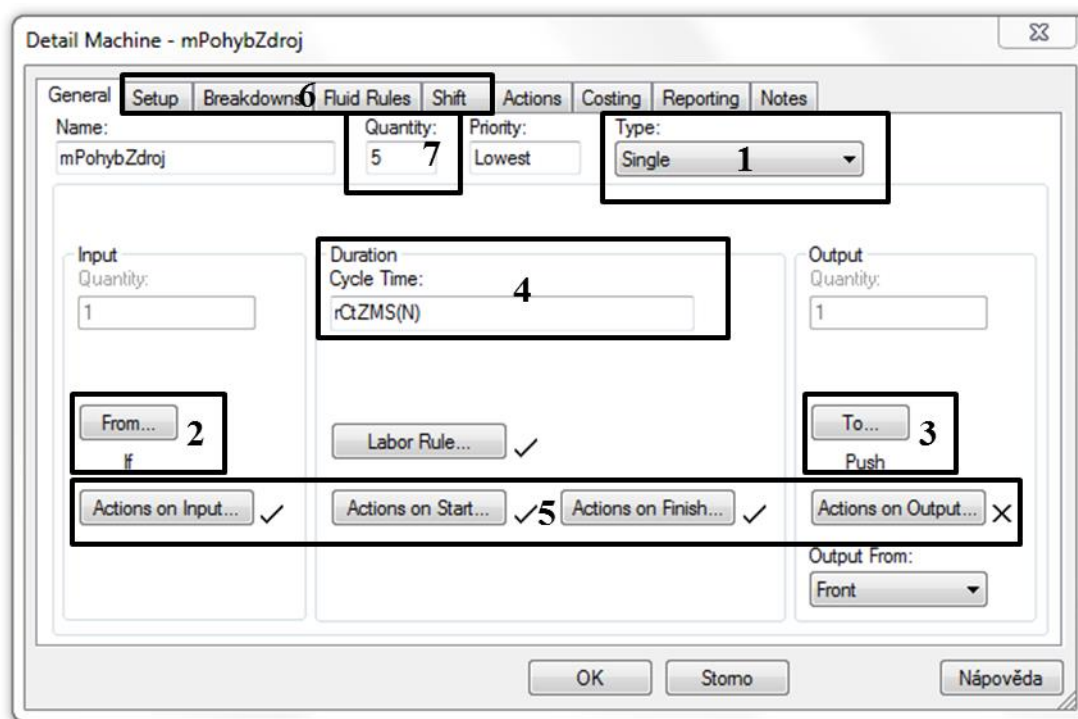
Prvek Part představuje entity, které procházejí modelem a jsou jím měněny. Tyto entity reprezentují kvantitu. Každý element typu Part musí mít definován svůj příchod do systému, který může být aktivní (Part přichází samostatně např. v pravidelných intervalech.), nebo pasivní, kdy je vstup entity do modelu realizován nějakým aktivním elementem (např. elementem Machine).

2. Buffer

Element Buffer slouží k modelování front. Jde o pasivní prvek, ve kterém čekají Parts na nějakou akci realizovanou aktivními elementy. Základními vlastnostmi Bufferu jsou jeho kapacita, doba zdržení entity a disciplína fronty. Disciplína fronty představuje způsob, jakým jsou entity řazeny do fronty po svém vstupu do Bufferu a jakým způsobem jsou vybírány aktivními elementy před opuštěním Bufferu.

3. Machine

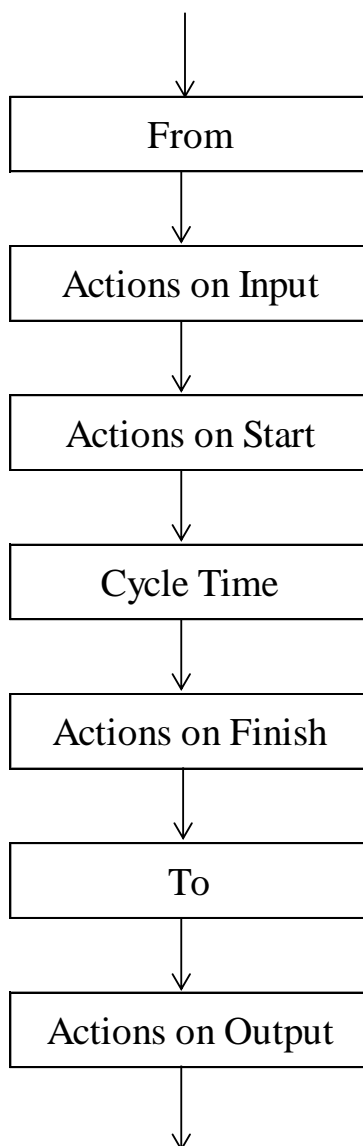
Machine je aktivní element, který pracuje s entitami. Disponuje celou řadou funkčních detailů, které lze nastavit v základním menu po jeho definování v modelu. Toto menu má podobu zachycenou na Obr. 5.3. Machine představuje klíčový element v každém modelu postaveném na principu DES. Zajišťuje totiž pohyb entit v simulaci, změnu jejich vlastností respektive změnu vlastností modelovaného systému.



Obr. 5.3 Detail elementu Machine po jeho definování v modelu

Důležitou vlastností elementu Machine je Type (viz 1 na Obr. 5.3). Tato vlastnost určuje počet vstupujících (Input – Quantity) a vystupujících (Output – Quantity) entit. Single Machine je nejčastější konfigurací, kdy počet vstupujících a vystupujících entit je roven 1. K dispozici je dále Assembly Machine s několika vstupujícími a 1 vystupující entitou, Production Machine s 1 vstupující entitou a několika vystupujícími a General Machine, kdy počet vstupujících a vystupujících entit je libovolný a zpravidla větší než 1. Entity jsou vtahovány do Machine pomocí vstupního pravidla (viz 2 na Obr. 5.3), přičemž nejčastěji používaným vstupním pravidlem je PULL. Pro každou entitu je dále specifikován čas, který stráví v elementu (viz 4 na Obr. 5.3) předtím, než pomocí výstupního pravidla (obvykle PUSH) element opouští (viz 3 na Obr. 5.3). Čas, který stráví entita v elementu Machine, se zadává buď přímo pomocí čísla, a to v případě, že se jedná o konstantu, či pomocí logického elementu Attribute nebo Variable. Základní tok entit elementem Machine, který lze popsat sekvencí vstupní pravidlo (FROM) – čas, který stráví entita v elementu (Cycle Time) – výstupní pravidlo (TO), lze dále rozšířit o akce (viz 5 na Obr. 5.3). Akce (Actions) představují programové struktury v jazyce VB, které slouží k nastavení hodnot logických elementů, a tím ke změně stavu systému, který tyto elementy

popisují. Pořadí pravidel, akcí a času, který stráví entita v elementu Machine, popisuje následující obrázek:



Obr. 5.4 Posloupnost akcí a pravidel, kterými prochází entita v elementu Machine

Pro programové struktury v akcích dále platí, že probíhají shora dolů po jednotlivých řádcích kódu. K simulaci vlivů, které způsobují přerušení práce elementu Machine, patří Setup, Breakdown a Shift (viz 6 na Obr. 5.3). Jedná se o funkční prvky, které se využívají k modelování seřízení, poruch a směnnosti. Zatímco Setup a Shift obvykle probíhají před operací vymezenou trváním Cycle Time elementu Machine, či po jejím skončení, Breakdown má náhodný

charakter, který se projevuje přerušením Cycle Time v jeho průběhu. Setup, Shift a Breakdown disponují jednak dobou trvání a také akcemi na začátku a na konci, které umožňují nastavení hodnot logických elementů a změnu stavu systému. Pro nastavení kvantity elementu Machine se využívá vlastnosti Quantity (viz 7 na Obr. 5.3). Jedná se o velmi důležitou vlastnost, která umožňuje například souběžné provádění stejných činností či souběžné využití více zdrojů na jednom elementu. Při stavbě modelu výrazně urychluje editaci programového kódu v akcích, neboť akce je nutné psát či upravovat pouze jednou. Na příslušný element o kvantitě větší než 1 se v tomto případě odkazuje pomocí písmene n .

4. Ostatní fyzické elementy

Labor představuje pracovníky nebo zdroje, které mohou být přidělovány strojům. V takovém případě může stroj pracovat, pouze je-li pracovník či zdroj k dispozici. Pracovníka je možné přiřadit jak celé operaci stroje, tak například pro opravu poruchy nebo seřízení. Element Vehicle se používá pro modelování manipulací pomocí manipulační techniky. Při použití tohoto prvku je třeba použít prvky typu Path, které definují cestu pro vozíky. Pro modelování pásových dopravníků slouží prvky typu Conveyor. Modules slouží k seskupování prvků modelu do přehledných struktur. Uplatní se zejména u rozsáhlejších modelů s komplikovanou hierarchickou strukturou.

Mezi logické elementy patří:

1. Attribute

Attribute reprezentuje nějakou vlastnost elementu Part. Jedná se o informaci, která může mít podobu celého či desetinného čísla, textu či jména nějakého elementu, který je již definován v modelu. Lze ho využít v akcích fyzických elementů pro provedení změny stavu systému a dále například pro nastavení Cycle Time či Output Quantity. Realizace akcí je vždy vázána na přítomnost Part.

2. Variable

Element Variable se využívá k popisu stavu systému a řízení logiky procesů v něm probíhajících. Stejně jako Part obsahuje informaci, která může mít podobu celého či desetinného čísla, textu či jména nějakého elementu, který je již definován v modelu. Velmi často se Variable vyskytuje v podobě vektoru, matice, kvádra či krychle, přičemž rozměry se udávají pomocí Quantity. Nejdříve se definuje počet sloupců, poté počet řádků a případně třetí rozměr. Element Variable je využíván v akcích fyzických elementů, ve vstupních a

výstupních pravidlech a dále například pro nastavení Cycle Time či Output Quantity. Dále lze Variable využít v Initialize Actions modelu pro načtení počátečních hodnot proměnných nejčastěji z Excelu (příkaz XLReadArray) a k nastavení struktury systému pomocí Quantity jednotlivých fyzických elementů (příkaz SET QUANTITY OF). Po skončení běhu simulace je možné hodnoty Variable zapsat zpět do Excelu pomocí User Actions modelu (příkaz XLWriteArray) nebo jsou hodnoty Variable využity v modulu Experimenter ke kalkulaci účelové funkce v případě, že se jedná o úlohu kombinatorické optimalizace.

3. Function

Element Function se nejčastěji využívá v modulu Experimenter ke kalkulaci účelové funkce na základě hodnot v elementech typu Variable. Dále je možné nahradit speciálním typem Function, který se nazývá Void programový kód v akcích fyzických elementů a akcích modelu. To je výhodné zejména tehdy, je-li programový kód velmi dlouhý, se složitým větvením a mnoha cykly.

4. Distribution

Distribution se používá ke generování náhodných čísel pravděpodobnostních rozdělení. K dispozici jsou rozdělení teoretická, u kterých je třeba definovat jejich parametry, a také rozdělení empirická, která vytváří uživatel přímo v modelu na základě sběru dat a jejich zpracování. Hodnoty elementu Distribution se obvykle zapisují do logických elementů Variable a Attribute a dále využívají v akcích fyzických elementů a akcích modelu.

Do skupiny grafických elementů patří Piechart a Timeseries. Jedná se o elementy, které při běhu simulace zobrazují vývoj logických elementů (Variable) v čase či dynamicky ukazují využití fyzických elementů (Machine, Labor, Vehicle) v simulovaném období.

Modul Experimenter (ve starších verzích Optimizer) nabízí algoritmy, které je možné využít při optimalizaci modelu. Nejpokročilejším optimalizačním algoritmem, který Witness nabízí, je Adaptive Thermostatical SA, pracující na principech algoritmu simulovaného žíhání. Dále jsou k dispozici algoritmus All Combinations, který provádí totální enumeraci všech možných řešení, algoritmus Min/Mid/Max, algoritmus Hill Climb, založený na lokálním prohledávání prostoru řešení, generátor náhodných řešení Random Solutions a algoritmus Six Sigma.

Modely vytvořené v prostředí Witness disponují možností 2D a v nejnovějších verzích také 3D grafického zobrazení modelovaného systému. Přímou v softwaru je k dispozici galerie obrázků, kterou lze dále rozšiřovat importem obrázků vlastních. To je výhodné například v případě, chceme-li v modelu použít 2D layout výrobní haly či skladu.

Witness také nabízí možnost spojitého modelování pomocí fyzických elementů, které jsou obdobou Part, Buffer a Machine. Jedná se o elementy Fluid, Tank a Processor.

V následující tabulce jsou uvedeny odkazy na literaturu, ve které se objevují některé aplikace simulačního nástroje Witness v modelování dodavatelských systémů.

Tabulka 5.4 Aplikace simulačního nástroje Witness v modelování dodavatelských systémů

Publikace	Stručný popis aplikace	Odvětví
Dyntar, et al. (2012)	Plánování a optimalizace skladovacích a stáčecích kapacit vstupů a finálních produktů při výrobě polybutadienu a styren-butadienu.	Chemický průmysl
Dyntar (2014)	Návrh systému manipulace dílů, polotovarů a hotových výrobků mezi skladem a výrobní linkou ve výrobě čerpadel.	Strojírenství
Abogrean (2014)	Plánování výrobních kapacit v cementárně.	Chemický průmysl
Dyntar and Strachotova (2015)	Návrh konceptu modelování materiálových toků s využitím agent-based paradigmatu v prostředí dynamické simulace, efektivní modelování systémů manipulace.	Neuvedeno
Strachotova and Pavlistik (2017)	Návrh systému zásobování milk-run při výrobě kabelových svazků do osobních automobilů.	Automobilový průmysl

6 NÁVRH OBECNÉHO SIMULAČNÍHO MODELU MATERIÁLOVÝCH TOKŮ PRO OPTIMALIZACI STRUKTURY DODAVATELSKÝCH SYSTÉMŮ

Modelování dodavatelských systémů je v mnoha ohledech velmi obtížné. Často se jedná o rozsáhlé modely různých procesů, které jsou řízeny různými strategiemi a které probíhají v organizacích s individuálními cíli. Pokud je takový model vytvořen v prostředí DES, stojí obvykle na základě rozsáhlé databáze vstupních dat často stochastické povahy. Tato vstupní data jsou získávána vlastním pozorováním a měřením, na základě rozhovorů s experty v organizacích, které dodavatelský systém tvoří, či exportem z informačních systémů těchto organizací. Sběr dat a jejich zpracování patří k časově nejnáročnějším etapám tvorby simulačního modelu a také k etapám, jejichž trvání značně kolísá a je pouze omezeně predikovatelné. To má za následek tlak na co možná nejrychlejší sestavení simulačního

modelu s co nejjednodušší strukturou, která zajistí jak rychlou validaci, tak maximální rychlost průběhu simulace při realizaci experimentů. Jednoduchá struktura simulačního modelu je ovšem v rozporu se samotnou složitostí a rozsahem modelovaného systému, neboť čím jednodušší a rychlejší model je, tím omezenější výstupy obvykle poskytuje.

6.1 Obecný simulační model materiálových toků vytvořený v prostředí Witness

Z předchozího popisu vyplývá, že požadavky při modelování materiálových toků v dodavatelských systémech pomocí simulace jsou následující:

- co nejrychlejší sestavení a validace modelu,
- co nejrychlejší běh modelu při experimentování a
- co nejširší paleta ukazatelů vhodných k posouzení výkonu modelovaného systému.

Výborným nástrojem, který umožňuje splnění výše uvedených požadavků, je simulační software Witness. Jak bylo popsáno v kapitole 5.1.3, Witness disponuje předdefinovanými elementy, jejichž typické vlastnosti je možné využít k efektivnímu modelování materiálových toků v dodavatelských systémech. Tyto elementy jsou využity k sestavení obecného simulačního modelu materiálových toků. Základní princip fungování modelu vychází z předpokladu, že jakýkoli materiálový tok lze rozložit do konečného počtu pohybů. Každý pohyb necht' představuje 3 po sobě následující akce, které spotřebovávají čas:

- manipulace s entitou v bodě 1,
- přemístění entity z bodu 1 do bodu 2,
- manipulace s entitou v bodě 2.

Příkladem takového pohybu je manipulace paletou z příjmové zóny skladu a její zaskladnění do pozice v regálu pomocí vysokozdvížného vozíku. Entita je v tomto případě 1 paleta, manipulace s entitou v bodě 1 je naložení palety v příjmové zóně skladu na vysokozdvížný vozík, přemístění entity z bodu 1 do bodu 2 je zajištěno prostřednictvím jízdy vysokozdvížného vozíku z příjmové zóny skladu do regálu a konečně manipulaci s entitou v bodě 2 představuje zaskladnění palety do pozice v regálu. Rozdělení na akce, ve kterých z pohledu simulace dochází a ve kterých nedochází ke změně polohy entity, je výhodné zejména při optimalizaci strategie řízení systému, neboť lze jasně kvantifikovat úspory, které generuje zkracování akcí spojených se změnou polohy entity. Nezbytnou

podmínkou pro uskutečnění pohybu je často využití nějakého zdroje. Mezi zdroje využívané v obecném simulačním modelu materiálových toků patří:

- stroje,
- vstupní materiál, polotovary, hotové výrobky a obaly,
- pracovníci,
- manipulační technika,
- skladovací kapacity a logistické plochy.

Zdroje využívané k realizaci pohybů disponují stavy, které jsou nositelem informací o jejich dostupnosti v každém okamžiku běhu simulace. Vzájemné interakce mezi zdroji pak přinášejí efekty, které je možné kvantifikovat pomocí ukazatelů a na základě jejich hodnot lze posoudit výkonnost modelovaného systému. Každý pohyb je realizován na základě požadavku. Jedná se o informaci, která jednak iniciuje samotné zahájení pohybu, jednak jednoznačně determinuje způsob jeho provedení. Požadavek specifikuje kvantitu, která prochází akcemi tvořícími pohyb, místa zahájení pohybu a jeho ukončení a odkaz na další informace týkající se například způsobu změny stavů zdrojů či délky trvání jednotlivých akcí.

Pohyb představuje základní prvek obecného simulačního modelu materiálových toků vytvořeného v prostředí Witness. Uvažujme například materiálový tok, který se dá rozložit do dvou pohybů (pohyb 1; pohyb 2). Pohyb 1 nechť představuje manipulaci s kompletovanou zakázkou z kompletační linky do expediční zóny skladu. Čas, kdy z kompletační linky sjedou 2 po sobě jdoucí zakázky, je náhodná veličina, kterou lze popsat rovnoměrným rozdělením s minimem 0,5 a maximem 3 minuty. Konec kompletační linky (bod 1) je vzdálen od středu expediční zóny skladu (bod 2) 83 metrů. K realizaci pohybu 1 je využit zdroj 1 reprezentovaný pracovníkem s elektrickým ručně vedeným vozíkem se zdvihem. Pracovník s vozíkem se pohybuje průměrnou rychlostí 5 km/h a při manipulaci se zakázkou v bodech 1 a 2 spotřebuje v průměru 0,5 minuty/zakázku a bod. Potřeba místa na uložení zakázky v expediční zóně skladu je náhodná veličina, kterou lze popsat rovnoměrným rozdělením s minimem 1 a maximem 4 m². Plocha expediční zóny skladu, do které lze ukládat kompletované zakázky, představuje zdroj 3 a je omezena na 100 m². Platí, že pokud nejsou zdroj 1 a zdroj 3 k dispozici, nelze provést pohyb 1. Pohyb 2 nechť představuje nakládku kompletovaných zakázek do kamionu přistaveného na expediční rampu (bod 3). Střed expediční zóny skladu (bod 2) je vzdálen od expediční rampy (bod 3) 56 metrů. K realizaci pohybu 2 je využit zdroj 2 reprezentovaný pracovníkem s ručně vedeným vozíkem bez zdvihu. Pracovník s vozíkem se pohybuje průměrnou rychlostí 3 km/h a při manipulaci se zakázkou v bodech 2 a 3 spotřebuje v průměru 0,6

minuty/zakázku, respektive 0,75 minuty/zakázku. Platí, že pokud není zdroj 2 k dispozici, nelze provést pohyb 2.

6.2 Struktura obecného simulačního modelu materiálových toků

Struktura obecného simulačního modelu materiálových toků pro simulaci materiálového toku, který je popsán v předchozí kapitole a který se skládá z výše uvedených pohybů, je zachycena na Obr. 6.1.

Samotná simulace je rozdělena na 6 bloků. Jedná se o:

- blok základní modelové struktury (ZMS),
- blok generování a přiřazování požadavků,
- blok stavy zdrojů,
- blok seznamy požadavků,
- blok informace o pohybech,
- blok výstupy.

Elementy použité v modelu včetně jejich Quantity jsou zobrazeny v části Element Tree. Jedná se o následující elementy:

pKulicka – element typu Part, který slouží k provedení Rules, Actions a Cycle Times veškerých elementů typu Machine

mCML – element typu Machine, který slouží k přiřazení požadavků na realizaci pohybu základní modelové struktury

mGeneratorPozPohyb1 – element typu Machine, který slouží k zápisu požadavků na provedení pohybu 1 do seznamu požadavků

mManipulaceBod1 – element typu Machine, který je součástí základní modelové struktury a slouží k simulaci manipulace s entitou v bodě 1 při realizaci pohybu

mManipulaceBod2 – element typu Machine, který je součástí základní modelové struktury a slouží k simulaci manipulace s entitou v bodě 2 při realizaci pohybu

mPohybBod1Bod2 – element typu Machine, který je součástí základní modelové struktury a slouží k simulaci přemístění entity z bodu 1 do bodu 2

mPohybZdroj – element typu Machine, který je součástí základní modelové struktury a slouží k přemístění zdroje do bodu 1, ve kterém má být zahájena realizace pohybu

mZapisVyuzitiZdroj – element typu Machine, který slouží k zapisování požadovaných výstupů do elementů typu Variable během simulace

IZdroj1 – element typu Labor, který slouží ke sledování využití zdrojů typu stroj, pracovník a manipulační technika

IZdroj2 – element typu Labor, který slouží ke sledování využití zdrojů typu stroj, pracovník a manipulační technika

iCisloZdroje – element typu Variable, který využívají elementy tvořící základní modelovou strukturu k rozpoznání pořadového čísla zdroje typu stroj, pracovník a manipulační technika

iKam – element typu Variable, který využívají elementy tvořící základní modelovou strukturu k rozpoznání směru pohybu

iKdejsemZdroj1 – element typu Variable s informací o aktuální poloze zdroje 1

iKdejsemZdroj2 – element typu Variable s informací o aktuální poloze zdroje 2

iKrok – element typu Variable využívaný v Actions elementu mZapisVyuzitiZdroj k zápisu požadovaných výstupů do elementů typu Variable během simulace

iOdkud – element typu Variable, který využívají elementy tvořící základní modelovou strukturu k rozpoznání směru pohybu

iPocPozSeznamPohyb1 – element typu Variable používaný k zápisu požadavků na provedení pohybu 1 do seznamu požadavků a při prohledávání tohoto seznamu požadavků s cílem přiřadit požadavek na provedení pohybu základní modelové struktury

iPocPozSeznamPohyb2 – element typu Variable používaný k zápisu požadavků na provedení pohybu 2 do seznamu požadavků a při prohledávání tohoto seznamu požadavků s cílem přiřadit požadavek na provedení pohybu základní modelové struktury

iPocZdroj1 – element typu Variable s informací o disponibilním počtu zdrojů typu stroj, pracovník a manipulační technika v modelu

iPocZdroj2 – element typu Variable s informací o disponibilním počtu zdrojů typu stroj, pracovník a manipulační technika v modelu

iStavZdroj1 – element typu Variable s informací o aktuálním stavu zdrojů typu stroj, pracovník a manipulační technika v modelu (tj. zda lze zdroj přiřadit na realizaci pohybu, či je zdroj při realizaci pohybu již využíván)

iStavZdroj12 – element typu Variable s informací o aktuálním stavu zdrojů typu stroj, pracovník a manipulační technika v modelu (tj. zda lze zdroj přiřadit na realizaci pohybu, či je zdroj při realizaci pohybu již využíván)

iStavZMS – element typu Variable s informací o aktuálním stavu základní modelové struktury v modelu (tj. zda lze základní modelové struktury přiřadit realizaci pohybu, či je základní modelová struktura při realizaci pohybu již využívána)

iVyuzitiZdroj1 – element typu Variable využívaný k zápisu požadovaných výstupů o využití zdroje typu stroj, pracovník a manipulační technika a jejich následný export

iVyuzitiZdroj2 – element typu Variable využívaný k zápisu požadovaných výstupů o využití zdroje typu stroj, pracovník a manipulační technika a jejich následný export

rCtZMS – element typu Variable využíváný k nastavení Cycle Times elementů tvořících základní modelovou strukturu

rMatvzdal – element typu Variable s informacemi o vzdálenostech bodů, mezi kterými se odehrávají pohyby (tj. matice vzdáleností)

rRezervaceZdroj3 – element typu Variable, který slouží k rezervaci zdroje typu vstupní materiál, polotovar, hotový výrobek, obal, skladovací kapacita a logistická plocha

rSeznamPozPohyb1 – element typu Variable používaný k zápisu požadavků na provedení pohybu 1 do seznamu požadavků a při prohledávání tohoto seznamu požadavků s cílem přiřadit požadavek na provedení pohybu základní modelové struktury

rSeznamPozPohyb2 – element typu Variable používaný k zápisu požadavků na provedení pohybu 2 do seznamu požadavků a při prohledávání tohoto seznamu požadavků s cílem přiřadit požadavek na provedení pohybu základní modelové struktury

rSpotrebaMista – element typu Variable, který využívají elementy tvořící základní modelovou strukturu k rozpoznání rezervovaného množství zdroje typu vstupní materiál, polotovar, hotový výrobek, obal, skladovací kapacita a logistická plocha spojeného s realizací pohybu

rStavZdroj3 – element typu Variable s informací o aktuálním disponibilním množství zdrojů typu vstupní materiál, polotovar, hotový výrobek, obal, skladovací kapacita a logistická plocha (tj. jaké maximální množství zdroje lze přiřadit na realizaci pohybu)

rTrvaniPohyby – element typu Variable s informací o trvání pohybů

rVyuzitiZdroj3 – element typu Variable využíváný k zápisu požadovaných výstupů o využití zdroje typu vstupní materiál, polotovar, hotový výrobek, obal, skladovací kapacita a logistická plocha a jejich následný export

sCesta – element typu Variable využíváný při realizaci Initialize Actions modelu k propojení simulace a souboru se vstupními daty v MS Excel

fPriradPozZMSPohyb1 – element typu Function využíváný k prohledávání seznamu požadavků na provedení pohybu s cílem přiřadit požadavek na provedení pohybu základní modelové struktury

fPriradPozZMSPohyb2 – element typu Function využíváný k prohledávání seznamu požadavků na provedení pohybu s cílem přiřadit požadavek na provedení pohybu základní modelové struktury

tZdroj3 – element typu Timeseries využíváný ke sledování využití zdroje typu vstupní materiál, polotovar, hotový výrobek, obal, skladovací kapacita a logistická plocha během simulace

pcZdroj1 – element typu Pie Chart využívaný ke sledování využití zdroje typu stroj, pracovník a manipulační technika během simulace

pcZdroj2 – element typu Pie Chart využívaný ke sledování využití zdroje typu stroj, pracovník a manipulační technika během simulace

Kromě těchto elementů, které jsou stálou součástí simulace, jsou dále v modelu využity elementy typu Variable, které vznikají a zanikají při realizaci Actions modelu, Actions elementů typu Machine a Function Bodies elementů typu Function. Jedná se o:

aa, bb – elementy typu Variable využívané v programových strukturách FOR/NEXT

mamzdroj1, mamzdroj2 – elementy typu Variable s informací o nalezení disponibilního zdroje typu stroj, pracovník a manipulační technika

mamzdroj3 – element typu Variable s informací o nalezení disponibilního zdroje typu vstupní materiál, polotovar, hotový výrobek, obal, skladovací kapacita a logistická plocha

mamZMS – element typu Variable s informací o nalezení disponibilní základní modelové struktury

Pro pojmenovávání elementů v simulaci platí, že první písmeno je malé a reprezentuje následující předdefinovaný element, který je v prostředí Witness k dispozici:

Tabulka 6.1 Pravidlo pro pojmenování elementů v obecném simulačním modelu materiálových toků vytvořeném v prostředí Witness

První písmeno	Element
p	Part
m	Machine
l	Labor
s	String Variable
i	Integer Variable
r	Real Variable
f	Function
pc	Pie Chart
t	Timeseries

Pojmenování elementů použitých v simulaci podle tohoto pravidla přispívá ke zrychlení stavby a validace modelu, neboť je na první pohled zřejmé, o jaký element se jedná, a je možné element okamžitě klonovat či upravovat jeho details. Při spuštění modelu dochází nejprve k načtení vstupních dat ze souboru vytvořeného v prostředí MS Excel, a to pomocí Initialize Actions modelu a příkazu XLReadArray, do elementů, které jsou součástí bloků stavy zdrojů a informace o pohybech. Kód inicializačních akcí modelu má následující podobu:

```
DIM aa AS INTEGER
```

```
!Nacti hodnoty do promennych
```

```
sCesta = "C:\\Users\\dyntarj\\Desktop\\Data.xlsx"
```

```
XLReadArray (sCesta,"Vstupy","B2",iPocZdroj1)
```

```
XLReadArray (sCesta,"Vstupy","B3",iPocZdroj2)
```

```
XLReadArray (sCesta,"Vstupy","B4",rStavZdroj3)
```

```
XLReadArray (sCesta,"Vstupy","B8:C10",rTrvaniPohyby)
```

```
XLReadArray (sCesta,"Vstupy","B14:D16",rMatvzdal)
```

```
!Nastav strukturu systemu
```

```
!Struktura ZMS
```

```
SET QUANTITY OF mPohybZdroj TO iPocZdroj1 + iPocZdroj2
```

```
SET QUANTITY OF mManipulaceBod1 TO iPocZdroj1 + iPocZdroj2
```

```
SET QUANTITY OF mPohybBod1Bod2 TO iPocZdroj1 + iPocZdroj2
```

```
SET QUANTITY OF mManipulaceBod2 TO iPocZdroj1 + iPocZdroj2
```

```
SET QUANTITY OF iStavZMS TO iPocZdroj1 + iPocZdroj2
```

```
SET QUANTITY OF rCtZMS TO iPocZdroj1 + iPocZdroj2
```

```
SET QUANTITY OF iOdkud TO iPocZdroj1 + iPocZdroj2
```

```
SET QUANTITY OF iKam TO iPocZdroj1 + iPocZdroj2
```

```
SET QUANTITY OF rSpotrebaMista TO iPocZdroj1 + iPocZdroj2
```

```
SET QUANTITY OF iCisloZdroje TO iPocZdroj1 + iPocZdroj2
```

!Struktura zdroje

!Zdroj1

SET QUANTITY OF iStavZdroj1 TO iPocZdroj1

SET QUANTITY OF iKdejsemZdroj1 TO iPocZdroj1

SET QUANTITY OF lZdroj1 TO iPocZdroj1

SET QUANTITY OF pcZdroj1 TO iPocZdroj1

!Zdroj2

SET QUANTITY OF iStavZdroj2 TO iPocZdroj2

SET QUANTITY OF iKdejsemZdroj2 TO iPocZdroj2

SET QUANTITY OF lZdroj2 TO iPocZdroj2

SET QUANTITY OF pcZdroj2 TO iPocZdroj2

!Nastav pocatecni polohu zdroju

!Zdroj1

FOR aa = 1 TO iPocZdroj1

 iKdejsemZdroj1(aa) = 3

NEXT

!Zdroj2

FOR aa = 1 TO iPocZdroj2

 iKdejsemZdroj2(aa) = 3

NEXT

Data v prostředí MS Excel mají podobu zachycenou na Obr. 6.2 a popisují množství zdrojů 1, 2, 3, které jsou k dispozici při běhu simulace, a v případě zdrojů 1 a 2 také rychlost jejich pohybu a čas spotřebovaný při manipulaci kompletovaných zakázek v rámci pohybů. Dále je součástí vstupních dat matice vzdáleností, která popisuje layout, v němž se pohyby tvořící simulovaný materiálový tok odehrávají. Pomocí příkazu SET QUANTITY OF je poté nastavena Quantity elementů tvořících blok ZMS a také elementů popisujících stavy zdrojů 1 a 2, které jsou součástí bloku stavy zdrojů. Quantity elementů tvořících ZMS

jsou nastaveny tak, aby bylo možné současně realizovat maximální množství pohybů s ohledem na množství zdrojů, které jsou k dispozici. Pro nastavení počáteční polohy zdrojů 1 a 2 (bod 3) je využito cyklu FOR/NEXT.

Microsoft Excel - Data

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	
1	Informace o zdrojích																			
2	Počet zdroj1	2																		
3	Počet zdroj2	3																		
4	Množství zdroj3 [m ³]	100																		
5																				
6	Informace o trvání pohybů																			
7	Charakteristika	Pohyb1	Pohyb2																	
8	Nakládka [min/pohyb]	0,5	0,6																	
9	Rychlost [km/hod]	5	3																	
10	Výkladka [min/pohyb]	0,5	0,75																	
11																				
12	Informace o layoutu - matice vzdáleností [m]																			
13	Odkud/Kam	1	2	3																
14	1	0	83	129																
15	2	83	0	56																
16	3	129	56	0																
17																				
18																				
19																				
20																				
21																				
22																				
23																				
24																				
25																				
26																				
27																				
28																				
29																				
30																				
31																				
32																				
33																				

Obr. 6.2 Vstupní data simulace v prostředí MS Excel

Simulace libovolného pohybu v modelu probíhá pomocí ZMS tvořené 4 elementy Machine Type Single, kterými prochází element Part pojmenovaný pKulicka v pořadí mPohybZdroj – mManipulaceBod1 – mPohybBod1Bod2 – mManipulaceBod2. Detaily těchto elementů jsou zachyceny na Obr. 6.3, Obr. 6.4, Obr. 6.5, Obr. 6.6. Machine mPohybZdroj je předřazen strojům simulujícím akce tvořící pohyb (viz kapitola 6.1) a slouží k modelování přemístění zdroje na místo, kde má být zahájena jeho realizace.

Práce elementů Machine tvořících ZMS je vázána na přiřazení požadavku na realizaci pohybu, který se projevuje změnou hodnoty elementu Variable iStavZMS. Platí, že stav 0 elementu iStavZMS znamená, že ZMS není přiřazen požadavek. Při přiřazení požadavku pak dochází ke změně stavové proměnné příslušné ZMS, kdy hodnota iStavZMS je nastavena na liché celé číslo větší než 0, přičemž platí, že každému pohybu je přiřazena právě 1 hodnota. Změna iStavZMS iniciuje činnost elementu mPohybZdroj a poté dochází k postupné realizaci Rules a Actions elementů typu Machine tvořících ZMS (viz kódy pod obrázky s detaily strojů tvořících ZMS). V Actions on Start jsou využívány informace z bloků stavy zdrojů a informace o pohybech k výpočtu Cycle Times a jejich nastavení pomocí Real Variable rCtZMS. V Actions on Finish elementů mPohybZdroj a mPohybBod1Bod2 je po dokončení Cycle Times měněna poloha zdrojů v závislosti na pohybu, který je simulován. V Actions on Output elementu mManipulaceBod1 je v případě

simulace pohybu 2 zvětšeno disponibilní množství zdroje 3 v závislosti na velikosti manipulované zakázky popsané v požadavku na realizaci pohybu. Klíčové z pohledu návaznosti pohybů tvořících simulovaný tok jsou Actions on Output elementu mManipulaceBod2. V případě simulace pohybu 1 jsou v těchto akcích zapisovány požadavky do seznamu požadavků na provedení pohybu 2, který má podobu Real Variable a je součástí bloku seznamy požadavků. V rámci těchto akcí dále dochází ke změně stavu zdrojů reprezentovaných elementy Integer a Real Variables. Zdroje typu stroj, pracovník a manipulační technika (v našem případě zdroj 1 a 2) disponují celočíselnými stavy, kdy stav 0 znamená, že zdroj je k dispozici, a stav větší než 0, že zdroj k dispozici není. Velmi užitečné je číslovat stavy zdrojů typu „není k dispozici“ vzestupně tak, že jedna číselná hodnota připadá vždy na 1 typ realizovaného pohybu. Jednoznačné přiřazení určité hodnoty stavu zdroji v každém okamžiku běhu simulace se využívá pro stanovení využití těchto zdrojů. Vrátime-li se do Actions on Output elementu mManipulaceBod2, potom změna stavů zdrojů 1 a 2, které jsou kódovány v elementech iStavZdroj1 a iStavZdroj2, na 0 znamená, že zdroje jsou k dispozici pro pokus o přiřazení dalšího požadavku na realizaci pohybu ze seznamu požadavků ZMS. Odlišné je pojetí stavů u zdrojů typu vstupní materiál, polotovar, hotový výrobek, obal skladovací kapacita a logistická plocha, tj. zdrojů, které jsou obvykle spojeny se zásobami. Stav 0 totiž znamená, že zdroj není k dispozici, a stavy větší než 0 udávají disponibilní množství zdroje v příslušných jednotkách. Často se jedná o spojitě veličiny nabývající neceločíselných hodnot, a proto jsou k jejich kódování ve větší míře využívány elementy Real Variables. Další rozdíl oproti zdrojům typu stroj, pracovník a manipulační technika je v okamžiku, kdy dochází ke změně stavu typu „není k dispozici“. Zatímco u prvně jmenovaných zdrojů dochází ke změně stavů typu „není k dispozici“ v okamžiku přiřazení požadavku ZMS (a tedy zahájení pohybu), u zdrojů spojených se zásobami dochází ke změně stavů až po dokončení pohybu. Důvodem je odstranění možného zkreslení průběhu stavu zásob či využití kapacity pro skladování v případě, že akce tvořící pohyb jsou časově náročné. Jako doplněk ke stavovým proměnným jsou proto v simulaci k dispozici ještě elementy typu Variable, které kódují tzv. rezervaci zdroje (viz rRezervaceZdroj3). Rezervované množství zdroje již není k dispozici při pokusu o přiřazení požadavku na realizaci pohybu ze seznamu požadavků ZMS, ale jeho skutečný stav reprezentovaný stavovou proměnnou se mění až po dokončení pohybu, tj. v Actions on Output elementu mManipulaceZdroj2.

Obr. 6.3 Detail elementu mPohybZdroj

Kód INPUT RULE elementu mPohybZdroj:

```

!Pozadavek na provedeni pohybu 1

IF iStavZMS(N) = 1

    PULL from pKulicka out of WORLD

!Pozadavek na provedeni pohybu 2

ELSEIF iStavZMS(N) = 3

    PULL from pKulicka out of WORLD

ELSE

    Wait

ENDIF

```

Kód LABOR RULE elementu mPohybZdroj:

!Prirad lZdroj pro sledovani vyuziti zdroju

!Pozadavek na provedeni pohybu 1

IF iStavZMS(N) = 2

lZdroj1(iCisloZdroje(N))

!Pozadavek na provedeni pohybu 2

ELSEIF iStavZMS(N) = 4

lZdroj2(iCisloZdroje(N))

ELSE

NONE

ENDIF

Kód OUTPUT RULE elementu mPohybZdroj:

PUSH to mManipulaceBod1(N)

Kód ACTIONS ON INPUT elementu mPohybZdroj:

!Zmen stav ZMS po provedeni vstupniho pravidla PULL pKulicka from WORLD

!Pozadavek na provedeni pohybu 1

IF iStavZMS(N) = 1

iStavZMS(N) = 2

!Pozadavek na provedeni pohybu 2

ELSEIF iStavZMS(N) = 3

iStavZMS(N) = 4

ENDIF

Kód ACTIONS ON START elementu mPohybZdroj:

!Nastav Cycle Time mPohybZdroj

!Pozadavek na provedeni pohybu 1

IF iStavZMS(N) = 2

$rCtZMS(N) = rMatvzdal(iKdejsemZdroj1(iCisloZdroje(N)), iOdkud(N)) / (rTrvaniPohyby(1, 2) * 1000 / 60)$

!Pozadavek na provedeni pohybu 2

ELSEIF iStavZMS(N) = 4

$rCtZMS(N) = rMatvzdal(iKdejsemZdroj2(iCisloZdroje(N)), iOdkud(N)) / (rTrvaniPohyby(2, 2) * 1000 / 60)$

ENDIF

Kód ACTIONS ON FINISH elementu mPohybZdroj:

!Zmen polohu zdroju

!Pozadavek na provedeni pohybu 1

IF iStavZMS(N) = 2

$iKdejsemZdroj1(iCisloZdroje(N)) = iOdkud(N)$

!Pozadavek na provedeni pohybu 2

ELSEIF iStavZMS(N) = 4

$iKdejsemZdroj2(iCisloZdroje(N)) = iOdkud(N)$

ENDIF

Obr. 6.4 Detail elementu mManipulaceBod1

Kód LABOR RULE elementu mManipulaceBod1:

!Prirad lZdroj pro sledovani vyuziti zdroju

!Pozadavek na provedeni pohybu 1

IF iStavZMS(N) = 2

lZdroj1(iCisloZdroje(N))

!Pozadavek na provedeni pohybu 2

ELSEIF iStavZMS(N) = 4

lZdroj2(iCisloZdroje(N))

ELSE

NONE

ENDIF

Kód OUTPUT RULE elementu mManipulaceBod1:

PUSH to mPohybBod1Bod2(N)

Kód ACTIONS ON START elementu mManipulaceBod1:

!Nastav Cycle Time mManipulaceBod1

!Pozadavek na provedeni pohybu 1

IF iStavZMS(N) = 2

$rCtZMS(N) = rTrvaniPohyby(1, 1)$

!Pozadavek na provedeni pohybu 2

ELSEIF iStavZMS(N) = 4

$rCtZMS(N) = rTrvaniPohyby(2, 1)$

ENDIF

Kód ACTIONS ON OUTPUT elementu mManipulaceBod1:

!Zmen stav zdroje

!Pozadavek na provedeni pohybu 2

IF iStavZMS(N) = 4

$rStavZdroj3 = rStavZdroj3 + rSpotrebaMista(N)$

ENDIF

Obr. 6.5 Detail elementu mPohybBod1Bod2

Kód LABOR RULE elementu mPohybBod1Bod2:

!Prirad lZdroj pro sledovani vyuziti zdroju

!Pozadavek na provedeni pohybu 1

IF iStavZMS(N) = 2

lZdroj1(iCisloZdroje(N))

!Pozadavek na provedeni pohybu 2

ELSEIF iStavZMS(N) = 4

lZdroj2(iCisloZdroje(N))

ELSE

NONE

ENDIF

Kód OUTPUT RULE elementu mPohybBod1Bod2:

PUSH to mManipulaceBod2(N)

Kód ACTIONS ON START elementu mPohybBod1Bod2:

!Nastav Cycle Time mPohybBod1Bod2

!Pozadavek na provedeni pohybu 1

IF iStavZMS(N) = 2

$rCtZMS(N) = rMatvzda(iOdkud(N), iKam(N)) / (rTrvaniPohyby(1, 2) * 1000 / 60)$

!Pozadavek na provedeni pohybu 2

ELSEIF iStavZMS(N) = 4

$rCtZMS(N) = rMatvzda(iOdkud(N), iKam(N)) / (rTrvaniPohyby(2, 2) * 1000 / 60)$

ENDIF

Kód ACTIONS ON FINISH elementu mPohybBod1Bod2:

!Zmen polohu zdroju

!Pozadavek na provedeni pohybu 1

IF iStavZMS(N) = 2

$iKdejsenZdroj1(iCisloZdroje(N)) = iKam(N)$

!Pozadavek na provedeni pohybu 2

ELSEIF iStavZMS(N) = 4

$iKdejsenZdroj2(iCisloZdroje(N)) = iKam(N)$

ENDIF

Detail Machine - mManipulaceBod2

General Setup Breakdowns Fluid Rules Shift Actions Costing Reporting Notes

Name: mManipulaceBod2 Quantity: 5 Priority: Lowest Type: Single

Input
 Quantity: 1
 From...
 Wait
 Actions on Input... ✕

Duration
 Cycle Time: rCtZMS(N)
 Labor Rule... ✓
 Actions on Start... ✓ Actions on Finish... ✕

Output
 Quantity: 1
 To...
 Push
 Actions on Output... ✓
 Output From: Front

OK Storno Nápověda

Obr. 6.6 Detail elementu mManipulaceBod2

Kód LABOR RULE elementu mManipulaceBod2:

!Prirad lZdroj pro sledovani vyuziti zdroju

!Pozadavek na provedeni pohybu 1

IF iStavZMS(N) = 2

lZdroj1(iCisloZdroje(N))

!Pozadavek na provedeni pohybu 2

ELSEIF iStavZMS(N) = 4

lZdroj2(iCisloZdroje(N))

ELSE

NONE

ENDIF

Kód OUTPUT RULE elementu mManipulaceBod2:

PUSH to SHIP

Kód ACTIONS ON START elementu mManipulaceBod2:

!Nastav Cycle Time mManipulaceBod2

!Pozadavek na provedeni pohybu 1

IF iStavZMS(N) = 2

$rCtZMS(N) = rTrvaniPohyby(1, 3)$

!Pozadavek na provedeni pohybu 2

ELSEIF iStavZMS(N) = 4

$rCtZMS(N) = rTrvaniPohyby(2, 3)$

ENDIF

Kód ACTIONS ON OUTPUT elementu mManipulaceBod2:

!Zmen stavy zdroju a ZMS a zapis pozadavek do seznamu

!Pozadavek na provedeni pohybu 1

IF iStavZMS(N) = 2

!Zapis pozadavku do seznamu

$iPocPozSeznamPohyb2 = iPocPozSeznamPohyb2 + 1$

$rSeznamPozPohyb2(1, iPocPozSeznamPohyb2) = 2$

$rSeznamPozPohyb2(2, iPocPozSeznamPohyb2) = 3$

$rSeznamPozPohyb2(3, iPocPozSeznamPohyb2) = rSpotrebaMista(N)$

!Zmen stavy zdroju a ZMS

$rStavZdroj3 = rStavZdroj3 - rSpotrebaMista(N)$

$rRezervaceZdroj3 = rRezervaceZdroj3 - rSpotrebaMista(N)$

$iStavZdroj1(iCisloZdroje(N)) = 0$

```

iStavZMS(N) = 0

!Pozadavek na provedeni pohybu 2

ELSEIF iStavZMS(N) = 4

!Zmen stavy zdroju a ZMS

iStavZdroj2(iCisloZdroje(N)) = 0

iStavZMS(N) = 0

ENDIF

!Vynuluj zbytek promennych z bloku ZMS

rCtZMS(N) = 0

iOdkud(N) = 0

iKam(N) = 0

rSpotrebaMista(N) = 0

iCisloZdroje(N) = 0

!Spust funkce, ktere prirazuji pozadavky ZMS

fPriradPozZMSPohyb1 ()

fPriradPozZMSPohyb2 ()

```

Jak bylo řečeno v předchozím textu, práce elementů Machine tvořících ZMS je vázána na přiřazení požadavku na realizaci pohybu, který je ZMS přiřazen ze seznamu požadavků. Seznam požadavků může být v simulaci realizován dvěma způsoby. Buď jsou požadavky shromažďovány v Bufferech v podobě Parts a nositelem informací o způsobu provedení pohybu jsou Attributes, nebo se požadavky kódují do elementů Real Variables jako čísla. Nevýhodou prvního přístupu je časté hromadění elementů Part v modelu, které způsobuje jeho zpomalování a také komplikovaný přístup ke všem požadavkům shromážděným v Bufferech. Proto je v simulaci využito kódování požadavků do elementů typu Real Variable (tj. rSeznamPozPohyb1 a rSeznamPozPohyb2), které tvoří blok seznamy požadavků. Platí, že každý řádek seznamu představuje 1 požadavek na realizaci pohybu a každý sloupec obsahuje informaci, která jednoznačně determinuje způsob provedení pohybu. V našem příkladě simulace materiálového toků složeného ze dvou pohybů je každý požadavek charakterizován 3 hodnotami, a to v jakém bodu pohyb začíná, v jakém

bodů pohyb končí a kolik místa v expediční zóně skladu manipulovaná zakázka zabere či uvolní. Proto proměnné rSeznamPozPohyb1 a rSeznamPozPohyb2 disponují 3 sloupci.

Zatímco požadavky na provedení pohybu 2 se zapisují do seznamu požadavků v Actions on Output elementu mManipulaceZdroj2 vždy po dokončení simulace pohybu 1, požadavky na provedení pohybu 1 zapisuje do seznamu element Machine, který se nazývá mGeneratorPozPohyb1. Detaily tohoto elementu jsou zachyceny na Obr. 6.7, kódy Rules a Actions elementu mGeneratorPozPohyb1 jsou uvedeny pod tímto obrázkem.

Obr. 6.7 Detail elementu mGeneratorPozPohyb1

Kód INPUT RULE elementu mGeneratorPozPohyb1:

PULL from pKulicka out of WORLD

Kód OUTPUT RULE elementu mGeneratorPozPohyb1:

PUSH to SHIP

Kód ACTIONS ON OUTPUT elementu mGeneratorPozPohyb1:

!Vygeneruj nový požadavek na realizaci pohybu

iPocPozSeznamPohyb1 = iPocPozSeznamPohyb1 + 1

!Odkud

rSeznamPozPohyb1(1, iPocPozSeznamPohyb1) = 1

!Kam

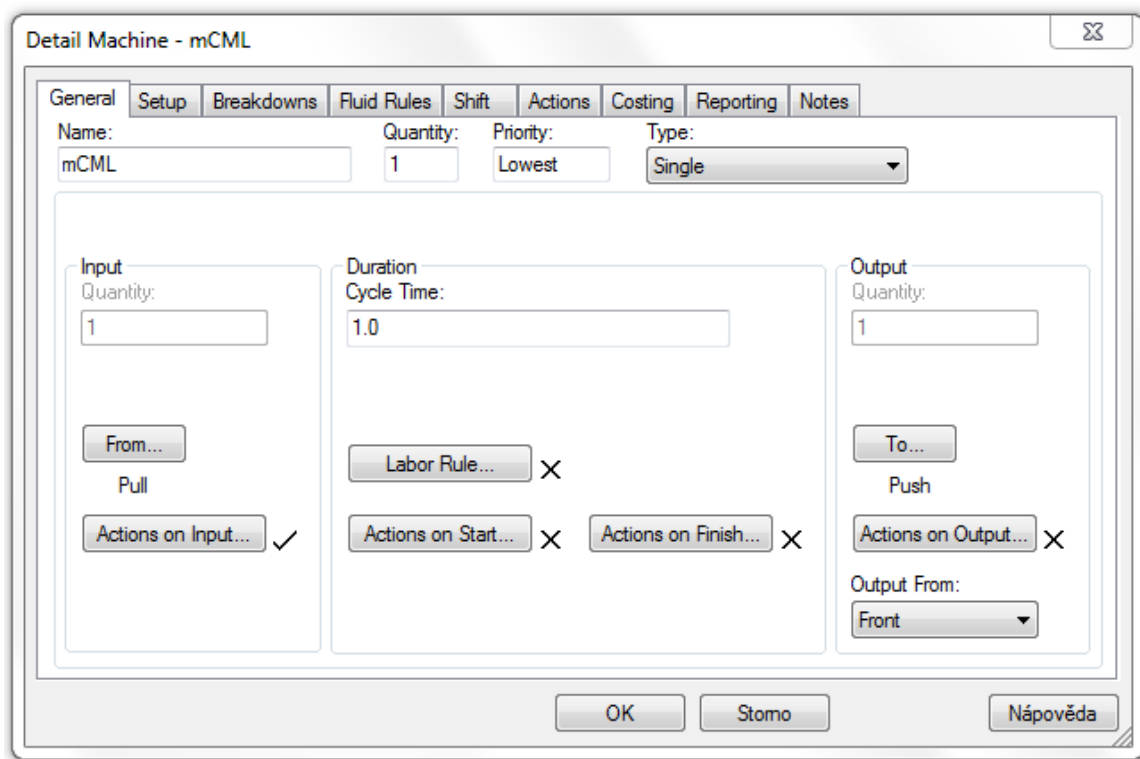
rSeznamPozPohyb1(2, iPocPozSeznamPohyb1) = 2

!Potreba Zdroj3

rSeznamPozPohyb1(3, iPocPozSeznamPohyb1) = Uniform (1,4)

Elementy Machine, které generují požadavky, jsou součástí bloku generování a přiřazování požadavků a v obecném simulačním modelu materiálových toků hrají velmi důležitou roli. Stojí totiž vždy na začátku sekvence na sebe navazujících pohybů, a tím účelně přerušují materiálový tok, který má být simulován. Používají se jednak ke generování požadavků na provedení pohybu, který stojí na začátku materiálového toku, a také ke generování požadavků v případě, že je z nějakého důvodu výhodnější nesimulovat všechny pohyby, do kterých lze tok rozložit. Typickým příkladem je výrobní linka, na které se provádí určitý počet výrobních operací vedoucích k produkci určitého množství hotového výrobku. V simulaci jsou výrobní operace nahrazeny generátorem požadavku na odvoz hotového výrobku v případě, že simulace pohybů představujících výrobní operace jsou zbytečné z pohledu cíle, kterého má být dosaženo. Simulace pohybů představujících materiálový tok skrz výrobní operace by v tomto případě pouze neúčelně zvyšovala rozsah modelu, a tím zpomalovala jeho běh.

Přiřazení požadavků na realizaci pohybu ZMS zajišťuje element Machine, který se nazývá mCML. Detaily tohoto elementu jsou zachyceny na Obr. 6.8, kódy Rules a Actions elementu mCML jsou uvedeny pod tímto obrázkem.



Obr. 6.8 Detail elementu mCML

Kód INPUT RULE elementu mCML:

PULL from pKulicka out of WORLD

Kód OUTPUT RULE elementu mCML:

PUSH to SHIP

Kód ACTIONS ON INPUT elementu mCML:

!Spust funkce, které prirazuji pozadavky ZMS

fPriradPozZMSPohyb1 ()

fPriradPozZMSPohyb2 ()

Element mCML využívá vstupního pravidla PULL pKulicka FROM WORLD, Actions on Input a výstupního pravidla PUSH TO SHIP k opakovanému prohledávání seznamů požadavků. Časté opakování je zajištěno krátkým Cycle Time elementu (obvykle jde o 1 časovou jednotku) a také neomezeným prováděním vstupního pravidla bez logických

podmínek. Pro každý požadavek zjišťuje mCML jednak dostupnost zdrojů nutných pro jeho realizaci a také dostupnost ZMS. Prohledávání seznamu požadavků je realizováno v Actions on Input cyklem se strukturou FOR/NEXT, zjišťování dostupnosti zdrojů a ZMS kombinací cyklů FOR/NEXT s logickými konstrukcemi IF/ELSEIF/ELSE/ENDIF naprogramovanými v jazyku VB. Vzhledem k tomu, že každý seznam požadavků reprezentuje 1 typ pohybu, který vyžaduje zapojení různých zdrojů, je každému prohledávání přiřazena unikátní programová struktura kombinující cykly a logické konstrukce. Aby kód ve VB popisující tyto programové struktury nebyl v Actions on Input mCML příliš rozsáhlý, je výhodné kód pro každý pohyb vložit do elementu Function typu Void. V našem příkladě se jedná o fPriradPozZMSPohyb1 a fPriradPozZMSPohyb2, jejichž Function Body popisuje kód s následující strukturou:

FUNCTION BODY funkce fPriradPozZMSPohyb1:

DIM aa AS INTEGER

DIM bb AS INTEGER

DIM mamzdroj1 AS INTEGER

DIM mamzdroj3 AS INTEGER

DIM mamZMS AS INTEGER

LABEL zacatek

!Pokud je na seznamu pozadavku nejaky pozadavek, najdi zdroje

IF iPocPozSeznamPohyb1 > 0

 mamzdroj1 = 0

 mamzdroj3 = 0

 mamZMS = 0

 FOR aa = 1 TO iPocPozSeznamPohyb1

 !Najdi zdroj1

 FOR bb = 1 TO iPocZdroj1

 IF iStavZdroj1(bb) = 0

 mamzdroj1 = bb

 GOTO dal1

```

ENDIF

NEXT

LABEL dal1

!Najdi zdroj 3

IF rStavZdroj3 - rRezervaceZdroj3 >= rSeznamPozPohyb1(3, aa)

    mamzdroj3 = aa

    GOTO dal2

ENDIF

LABEL dal2

!Najdi volnou ZMS

FOR bb = 1 TO iPocZdroj1 + iPocZdroj2

    IF iStavZMS(bb) = 0

        mamZMS = bb

        GOTO dal3

    ENDIF

NEXT

LABEL dal3

!Pokud jsou k dispozici zdroje a ZMS

IF mamzdroj1 > 0 AND mamzdroj3 > 0 AND mamZMS > 0

!Zadej pozadavek ZMS

    iStavZMS(mamZMS) = 1

    iOdkud(mamZMS) = rSeznamPozPohyb1(1, aa)

    iKam(mamZMS) = rSeznamPozPohyb1(2, aa)

    rSpotrebaMista(mamZMS) = rSeznamPozPohyb1(3, aa)

    iCisloZdroje(mamZMS) = mamzdroj1

```

!Zmen stavy zdroju

iStavZdroj1(mamzdroj1) = 1

rRezervaceZdroj3 = rRezervaceZdroj3 + rSeznamPozPohyb1(3, aa)

!Odeber pozadavek z seznamu

FOR bb = aa TO iPocPozSeznamPohyb1

rSeznamPozPohyb1(1, bb) = rSeznamPozPohyb1(1, bb + 1)

rSeznamPozPohyb1(2, bb) = rSeznamPozPohyb1(2, bb + 1)

rSeznamPozPohyb1(3, bb) = rSeznamPozPohyb1(3, bb + 1)

NEXT

iPocPozSeznamPohyb1 = iPocPozSeznamPohyb1 - 1

GOTO zacatek

ENDIF

NEXT

ENDIF

FUNCTION BODY funkce fPriradPozZMSPohyb2:

DIM aa AS INTEGER

DIM bb AS INTEGER

DIM mamzdroj2 AS INTEGER

DIM mamZMS AS INTEGER

LABEL zacatek

!Pokud je na seznamu pozadavku nejaky pozadavek, najdi zdroje

IF iPocPozSeznamPohyb2 > 0

mamzdroj2 = 0

mamZMS = 0

FOR aa = 1 TO iPocPozSeznamPohyb2

!Najdi zdroj2

```
FOR bb = 1 TO iPocZdroj2
    IF iStavZdroj2(bb) = 0
        mamzdroj2 = bb
        GOTO dal1
    ENDIF
NEXT
LABEL dal1
```

!Najdi volnou ZMS

```
FOR bb = 1 TO iPocZdroj1 + iPocZdroj2
    IF iStavZMS(bb) = 0
        mamZMS = bb
        GOTO dal2
    ENDIF
NEXT
LABEL dal2
```

!Pokud jsou k dispozici zdroje a ZMS

```
IF mamzdroj2 > 0 AND mamZMS > 0
```

!Zadej pozadavek ZMS

```
iStavZMS(mamZMS) = 3
iOdkud(mamZMS) = rSeznamPozPohyb2(1, aa)
iKam(mamZMS) = rSeznamPozPohyb2(2, aa)
rSpotrebaMista(mamZMS) = rSeznamPozPohyb2(3, aa)
iCisloZdroje(mamZMS) = mamzdroj2
```

!Zmen stavy zdroju

```

iStavZdroj2(mamzdroj2) = 1

!Odeber pozadavek z seznamu

FOR bb = aa TO iPocPozSeznamPohyb2

    rSeznamPozPohyb2(1, bb) = rSeznamPozPohyb2(1, bb + 1)

    rSeznamPozPohyb2(2, bb) = rSeznamPozPohyb2(2, bb + 1)

    rSeznamPozPohyb2(3, bb) = rSeznamPozPohyb2(3, bb + 1)

NEXT

iPocPozSeznamPohyb2 = iPocPozSeznamPohyb2 - 1

GOTO zacatek

ENDIF

NEXT

ENDIF

```

V Actions on Input mCML se pak uvádí seznam těchto elementů v pořadí, ve kterém se mají prohledávat seznamy požadavků. Dostupnost zdrojů a ZMS je posuzována na základě hodnot elementů typu Variable, které jsou součástí bloku stavu zdrojů a elementu iStavZMS. V případě, že lze požadavek realizovat, je přiřazen ZMS, odstraněn ze seznamu požadavků a je provedena změna stavů zdrojů a ZMS vybraných k jeho provedení. Tím, že dochází k dynamickému zkracování seznamů při přiřazení požadavku ZMS, je jejich prohledávání pomocí cyklu FOR/NEXT maximálně efektivní. Navíc lze toto prohledávání také řídit v případě, že podmínkou je například dodržení strategie first in first out či nějaký požadavek na provedení pohybu disponuje prioritou. Stejně tak využití příkazu GOTO a Labels v cyklech FOR/NEXT umožňuje maximální efektivitu při hledání zdrojů, které je třeba využít při realizaci pohybu. Aby nedocházelo k prodlevě mezi okamžikem, kdy dochází k uvolňování zdrojů při dokončení pohybu, a Actions on Input elementu mCML, jsou elementy Functions, které přiřazují požadavky na provedení pohybu ZMS, využity také v Actions on Output elementu mManipulaceBod2.

Výstupy simulace jsou realizovány 2 způsoby. Elementy Pie Charts a Timeseries umožňují sledovat využití zdrojů přímo při běhu simulace, a používají se proto při prezentaci grafiky modelu. V našem případě jsou pro sledování využití zdrojů typu stroj, pracovník a manipulační technika (tj. zdroj 1 a zdroj 2) použity elementy typu Labor (lZdroj1 a lZdroj2), které jsou pomocí Labor Rules přiřazovány elementům Machines, jež tvoří ZMS. Příkazem SUtil je v průběhu simulace zjišťováno využití elementů lZdroj1 a

lZdroj2, které je opakovaně zobrazováno v elementech typu Pie Chart s názvem pcZdroj1 a pcZdroj2 (viz Obr. 6.9 a Obr. 6.10).

Detail Pie Chart - pcZdroj1

General | Element States | Actions | Reporting | Notes

Name: pcZdroj1 Quantity: 2 Refresh Interval: 1.0

Sectors

	Description	Expression
1	Busy	SUtil (lZdroj1(N),2)
2	Idle	SUtil (lZdroj1(N),1)

Actions

Before Refresh... X After Refresh... X

Shift: Undefined

Pull Out Sector:

WPM

☐ Broadcast

OK Storno Nápověda

Obr. 6.9 Detail elementu pcZdroj1 pro grafické zobrazení využití zdrojů

Detail Pie Chart - pcZdroj2

General | Element States | Actions | Reporting | Notes

Name: pcZdroj2 Quantity: 3 Refresh Interval: 1.0

Sectors

	Description	Expression
1	Busy	SUtil (lZdroj2(N),2)
2	Idle	SUtil (lZdroj2(N),1)

Actions

Before Refresh... X After Refresh... X

Shift: Undefined

Pull Out Sector:

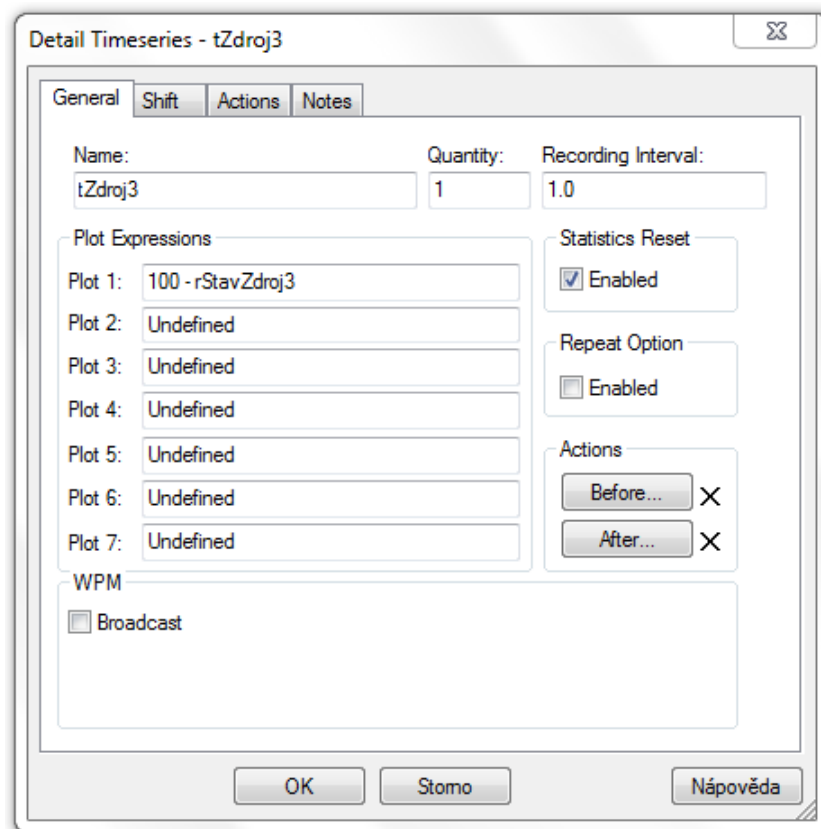
WPM

☐ Broadcast

OK Storno Nápověda

Obr. 6.10 Detail element pcZdroj2 pro grafické zobrazení využití zdrojů

Pro zdroje typu vstupní materiál, polotovary, hotový výrobek, obal, skladovací kapacita a logistická plocha (tj. zdroj 3) je ke sledování využití během simulace využit element Timeseries s názvem tZdroj3, který opakovaně zobrazuje využití expediční plochy skladu pomocí elementu rStavZdroj3 (viz Obr. 6.11).



Obr. 6.11 Detail elementů tZdroj3 pro grafické zobrazení využití zdrojů

Často je ovšem užitečné zapisovat během simulace výstupy do elementů Variables a ty poté exportovat do MS Excel pomocí User Actions modelu. Jde například o situaci, kdy u zdrojů typu stroj, pracovník a manipulační technika nestačí sledovat jejich využití v podobě pracuje/nepracuje, ale zajímá nás, kolik času zdroje spotřebovaly na konkrétní činnosti. Dalším příkladem může být sledování zásob zdrojů typu vstupní materiál, polotovary, hotový výrobek, kdy po exportu stavů zásob zaznamenaných během simulace do MS Excel jsou prováděny další analýzy spojené například s výpočtem doby obratu. K tomu je v simulaci využit element Machine s názvem mZapisVyuzitiZdroj a sada Variables, do kterých jsou zapisovány požadované hodnoty. Details elementu mZapisVyuzitiZdroj jsou na Obr. 6.12, kódy pro Rules a Actions elementu mZapisVyuzitiZdroj jsou uvedeny pod tímto obrázkem.

Obr. 6.12 Detail elementu mZapisVyuzitiZdroj

Kód INPUT RULE elementu mZapisVyuzitiZdroj:

PULL from pKulicka out of WORLD

Kód OUTPUT RULE elementu mZapisVyuzitiZdroj:

PUSH to SHIP

Kód ACTIONS ON FINISH elementu mZapisVyuzitiZdroj:

DIM aa AS INTEGER

!Zapis stav zdroje

iKrok = iKrok + 1

!Zdroj1

FOR aa = 1 TO iPocZdroj1

iVyuzitiZdroj1(aa, iKrok) = iStavZdroj1(aa)

NEXT

!Zdroj2

FOR aa = 1 TO iPocZdroj2

iVyuzitiZdroj2(aa, iKrok) = iStavZdroj2(aa)

NEXT

!Zdroj3

rVyuzitiZdroj3(iKrok) = 100 - rStavZdroj3

User Actions modelu pro export výstupů do MS Excel pomocí příkazu XLWriteArray jsou popsány pomocí následujícího kódu:

Kód USER ACTIONS modelu:

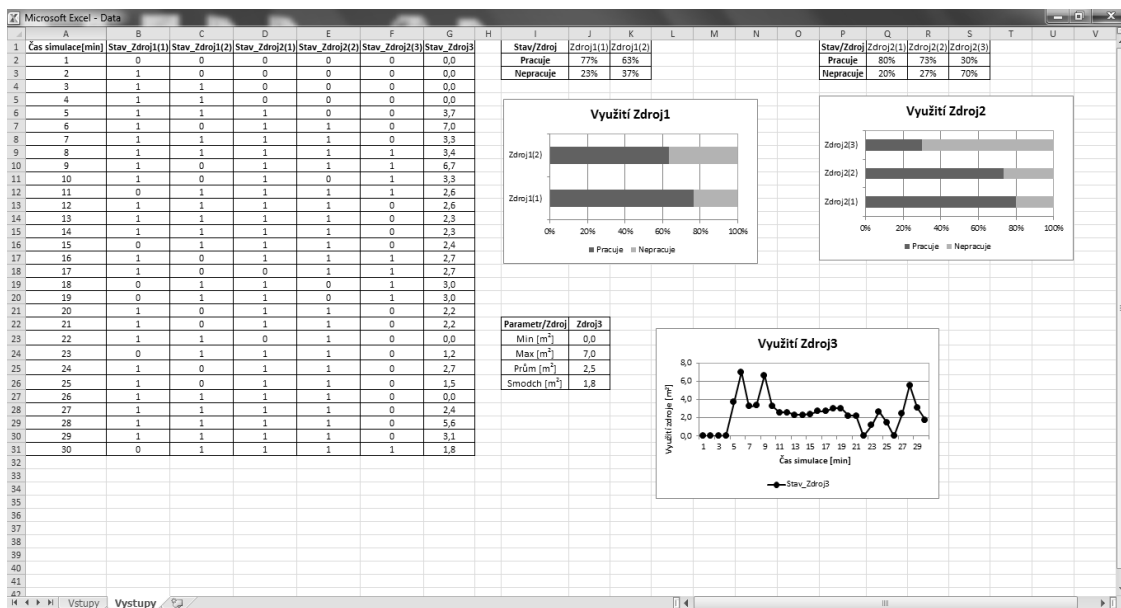
!Zapis stavy zdroju do MS Excel

XLWriteArray (sCesta,"Vystupy","B2:C31",iVyuzitiZdroj1)

XLWriteArray (sCesta,"Vystupy","D2:F31",iVyuzitiZdroj2)

XLWriteArray (sCesta,"Vystupy","G2:G31",rVyuzitiZdroj3)

Tyto výstupy jsou poté zpracovány pomocí matematických a statistických funkcí a jsou upraveny do požadované grafické podoby (viz. Obr. 6.13).



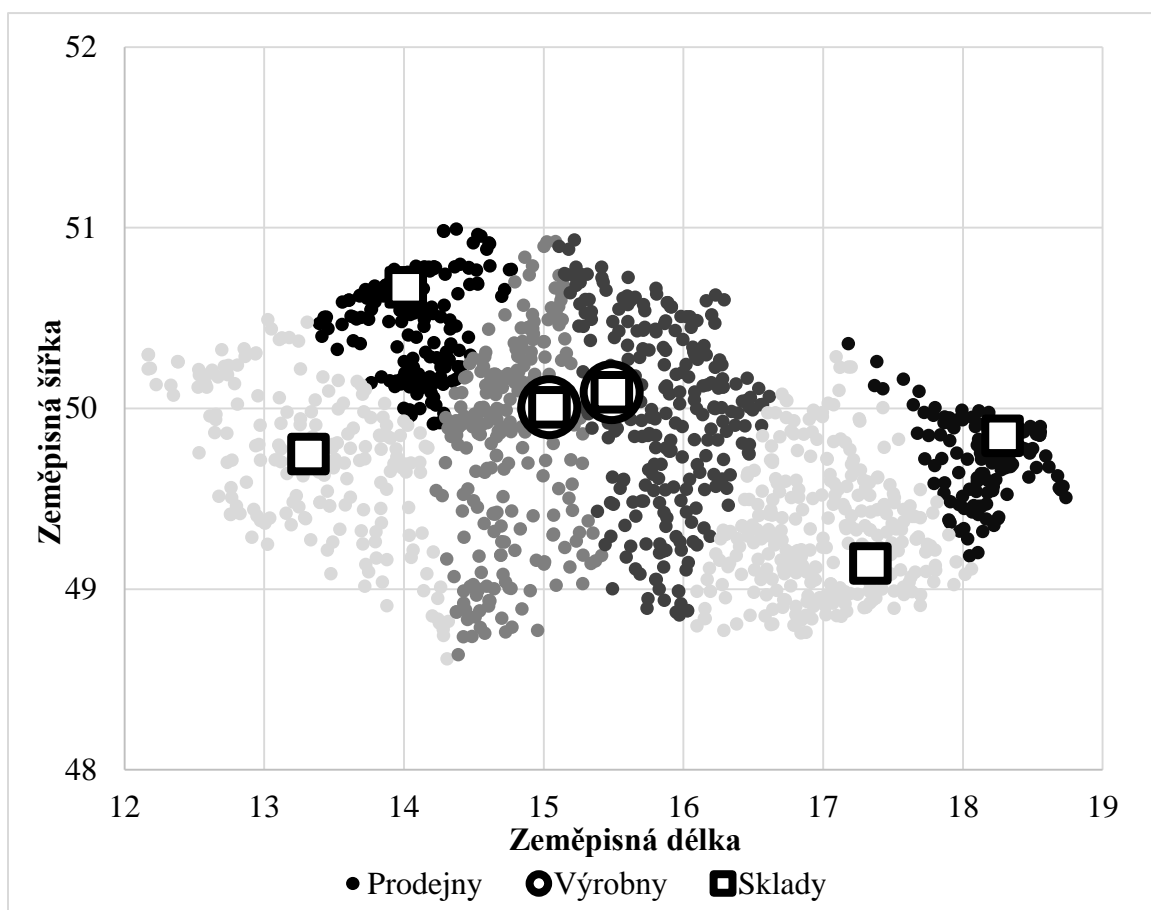
Obr. 6.13 Výstupy simulace v prostředí MS Excel

7 PŘÍKLADY APLIKACÍ OBECNÉHO SIMULAČNÍHO MODELU MATERIÁLOVÝCH TOKŮ

7.1 Redesign distribučního systému společnosti zabývající se výrobou a distribucí olejů a maziv

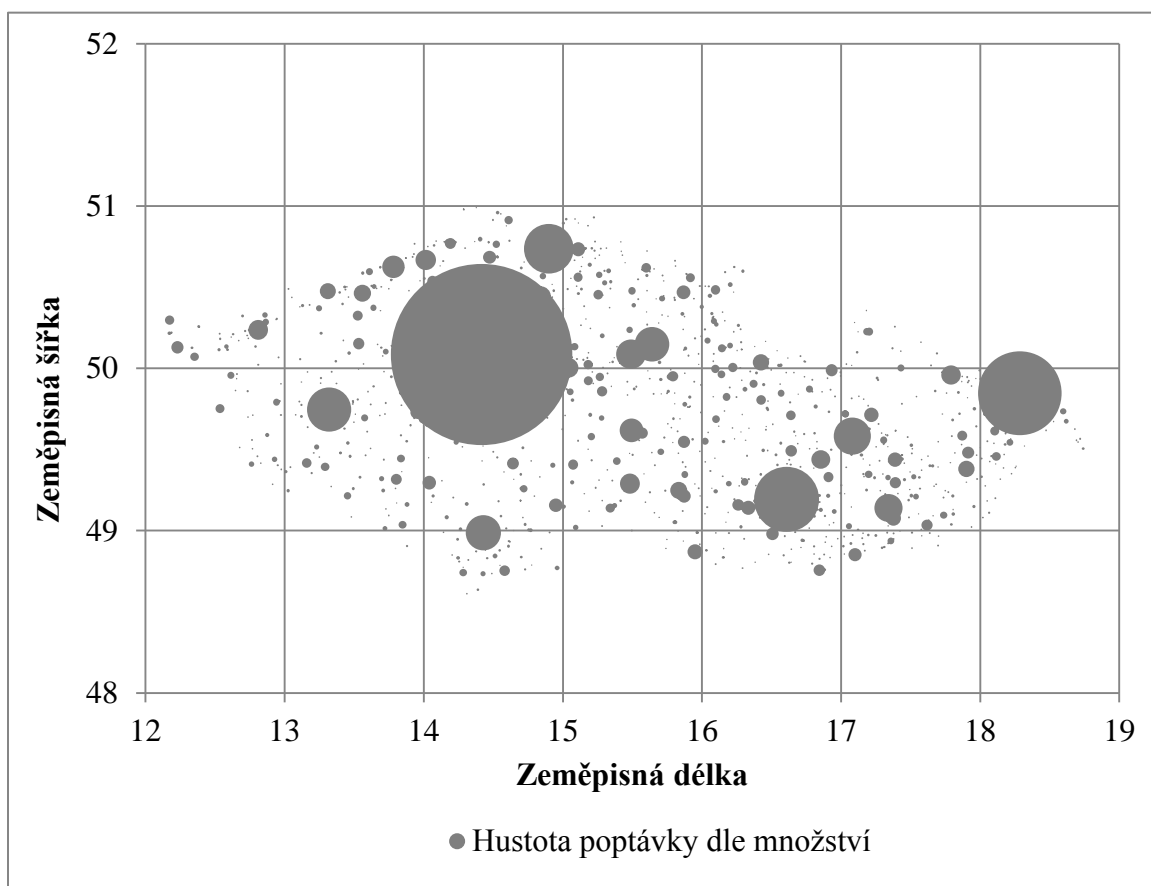
7.1.1 Současný stav struktury distribučního systému

Produkty v modelovaném systému jsou v současnosti vyráběny ve 2 výrobních závodech ležících na území České republiky. Jedná se o oleje a maziva plněná do plastových obalů, skladovaná a manipulovaná na paletách. Výrobní kapacity závodů jsou dostatečné k uspokojení veškeré poptávky konečných zákazníků. Celkové produkované množství výrobků se pohybuje okolo 30 000 tun za rok. Výrobky jsou prodávány v síti 1 300 obchodů umístěných na celém území České republiky (viz Obr. 7.1).



Obr. 7.1 Současná struktura distribučního systému

Prodejny jsou zásobovány z 6 skladů s termínem vyřízení objednávky 2 dny a požadovanou hodnotou úrovně služeb ve výši 98 %. Úroveň služeb je stanovena jako podíl množství dodaného ze skladové zásoby a celkového objednaného množství.⁷ Příslušnost prodejny do teritoria skladu je stanovena na základě minimální vzdálenosti mezi prodejnou a všemi sklady. Rozložení a hustota poptávky konečných zákazníků, kteří nakupují v prodejnách, jsou zachyceny na Obr. 7.2.

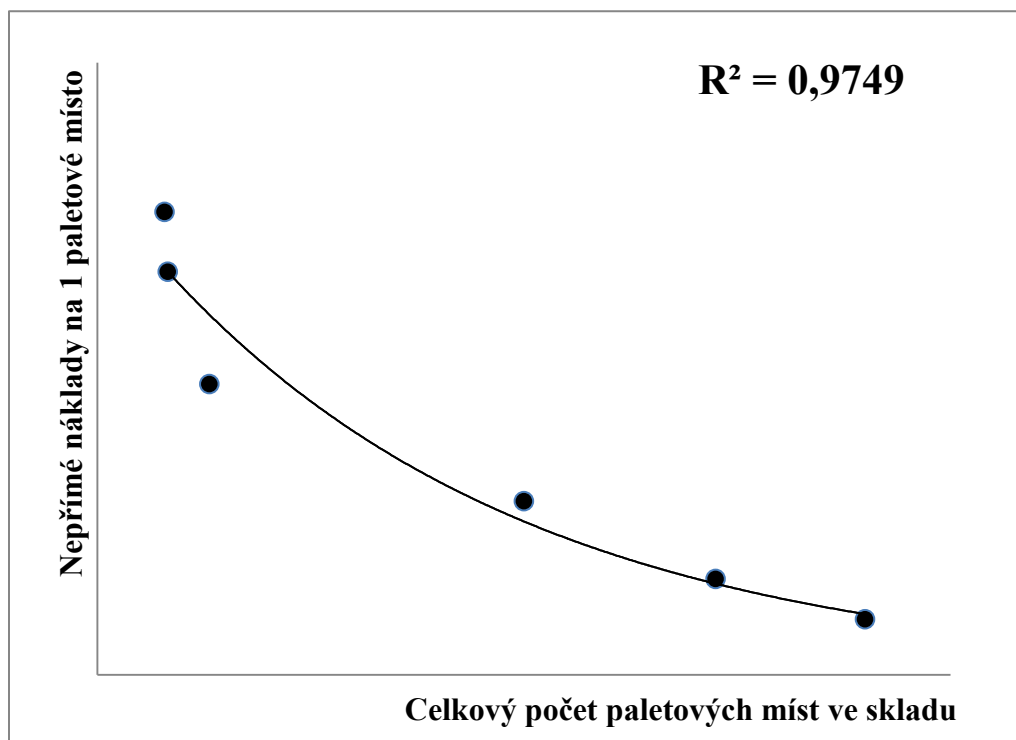


Obr. 7.2 Rozložení poptávky a její hustota

Největší poptávka po produktech je v Praze a v jejím okolí a také v krajských městech. Takové rozložení poptávky a její hustota jsou, v případě České republiky typické zejména pro distribuci spotřebního zboží. Výrobny jsou schopny doplňovat sklady požadovaným sortimentem výrobků v horizontu dvou týdnů. Kapacity dopravních prostředků využívané v distribučním systému se pohybují v rozmezí 1–7 tun/kamion se sazbami účtovanými dopravci v rozmezí 0,345–0,711 €/km. Skladovací kapacity ve skladech se pohybují od 600 do 7 200 paletových míst při trendu chování ročních nepřímých nákladů, který je zachycen

⁷ Tj. ukazatel fill rate.

na Obr. 7.3. Nepřímé náklady zahrnují především poplatky za údržbu informačního systému, poplatky za údržbu systému řízení skladu a náklady na energie spotřebované při provozu skladu.



Obr. 7.3 Provozní náklady skladů

Diskontní sazba zohledňující míru vázanosti kapitálu v zásobách je uvažována ve výši 1,53 %. Náklady na 1 pracovníka skladu jsou 1 000 €/měsíc a náklady spojené s pronájmem elektrických ručně vedených vozíků se zdvihem používaných k manipulaci ve skladech jsou 6 800 €/vozík, rok.

7.1.2 Modelování distribučního systému

Model distribučního systému je realizován ve 3 navazujících fázích. V první fázi je provedena lokalizace vybraného počtu skladů v závislosti na poptávaném množství a geografickém umístění odběratelů. K tomu slouží Lokalizační algoritmus (LA), který je realizován v prostředí MS Excel. LA v tomto případě reprezentuje algoritmus, který zajišťuje návrh struktury systému. Na základě výstupů LA je ve druhé fázi navržen způsob, jakým jsou řízeny zásoby ve skladech, výše potřebné skladovací kapacity a způsob, jakým jsou generovány doplňující objednávky. Tato fáze je realizována pomocí algoritmu Simulace pohybů skladové zásoby (PSMS) rovněž v prostředí MS Excel. Výstupy první a druhé fáze jsou poté importovány do prostředí simulačního softwaru Witness a je

provedena simulace materiálových toků s cílem posouzení ekonomické efektivity navržené struktury distribučního systému. K plánování rozvožových tras mezi sklady a odběrateli je využit Clarke&Wright's Savings algoritmus (CWS).

Fáze 1: Lokalizace skladů pomocí LA

Cílem této fáze je nalézt optimální umístění vybraného počtu skladů, kteří obsluhují skupinu prodejen. K tomu slouží Lokalizační algoritmus (LA). LA je vytvořen v prostředí MS Excel.

Vstupní data do LA

- Global Position System (GPS) – souřadnice obsluhovaných zákazníků (tj. prodejen),
- celkové poptávané množství produktů v určitém časovém období (např. za rok),
- počet lokalizovaných objektů (tj. skladů).

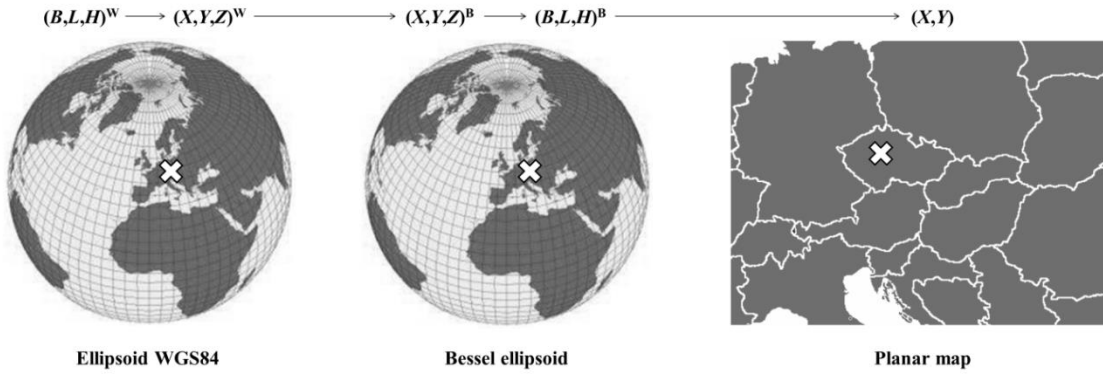
Princip fungování LA

LA je definován jako lokalizace $i = 1, 2, \dots, m$ nových objektů v síti $j = 1, 2, \dots, n$ existujících zákazníků. Základním předpokladem je, že každý objekt může obsluhovat každého zákazníka a každý zákazník je obsluhován pouze jedním logistickým objektem. Cílem lokalizace je umístit objekty tak, aby bylo dosaženo minimalizace účelové funkce Z v podobě:

$$Z = \sum_{i=1}^m \sum_{j=1}^n q_{ij} d_{ij} \quad (7.1)$$

kde d_{ij} je vzdálenost a q_{ij} je celkové dopravované množství mezi objekty a zákazníky v určitém časovém období. Zatímco množství dopravovaných produktů je specifikováno ve vstupních datech, vzdálenosti mezi objekty a zákazníky musí být získány výpočtem jako vzdálenosti mezi hledanou lokací nových objektů o souřadnicích (X_i, Y_i) a lokací obsluhovaných zákazníků o souřadnicích (X_j, Y_j) . V prvním kroku jsou GPS souřadnice zákazníků transformovány do roviny tak, aby vzdálenosti korespondovaly se skutečností. Vzhledem k tomu, že distribuční systém se nachází na území ČR, k transformaci GPS souřadnic je použit systém navržený pro střední Evropu, jehož základní princip je zachycen na Obr. 7.4. Vstupní hodnoty GPS souřadnic, tj. severní šířka $(B)^W$ a východní délka $(L)^W$, jsou nejprve transformovány do pravoúhlých souřadnic $(X, Y, Z)^W$ v geocentrickém systému elipsoidu WGS84. Ačkoli hodnota nadmořské výšky $(H)^W$ nemá vliv na lokaci objektu v rovině, je nutné použít její hodnotu v prvních třech krocích transformace. Na počátku je její hodnota rovna nule. V dalším kroku transformace je provedena transformace na Besselův elipsoid $(X, Y, Z)^B$ a zpětná transformace z pravoúhlých do geodetických

souřadnic $(B, L, H)^B$. V posledním kroku jsou spočteny pravoúhlé souřadnice (X, Y) . Systém transformovaných lokací je navržen v metrech.



Obr. 7.4 Ilustrace transformace GPS souřadnic do roviny

Na základě transformovaných souřadnic jsou spočteny vzdálenosti d_{ij} pomocí rovnice:

$$d_{ij} = k_s \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2} \quad (7.2)$$

kde k_s je korekce přímé vzdálenosti závislá například na hustotě silniční sítě. V LA je použita hodnota $k_s = 1,235$ [viz Škvor, et al. (2011)]. Optimální poloha objektů v síti zákazníků je poté zjištěna minimalizací účelové funkce v podobě rovnice 7.1. V případě, že je počet lokalizovaných objektů roven 1, stačí derivace účelové funkce podle X a Y položit rovné nule a po dalších úpravách dostaneme výrazy v podobě rovnic:

$$X \sum_{j=1}^n \frac{q_j}{\sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}} = \sum_{j=1}^n \frac{q_j X_j}{\sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}} \quad (7.3)$$

$$Y \sum_{j=1}^n \frac{q_j}{\sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}} = \sum_{j=1}^n \frac{q_j Y_j}{\sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}} \quad (7.4)$$

Jelikož není možné přímo vyjádřit hodnoty souřadnic X a Y , je dále použita iterativní metoda s využitím pomocné funkce v podobě rovnice:

$$f_j = \frac{q_j}{\sqrt{(X - X_j)^2 + (Y - Y_j)^2} + \varepsilon} \quad (7.5)$$

kde ε je hodnota blízká nule (např. 10^{-7}) která vylučuje dělení nulou v situaci kdy $X = X_j$ a $Y = Y_j$. Konečné souřadnice je poté možno získat jako:

$$X = \frac{\sum_{j=1}^n f_j X_j}{\sum_{j=1}^n f_j} \quad (7.6)$$

$$Y = \frac{\sum_{j=1}^n f_j Y_j}{\sum_{j=1}^n f_j} \quad (7.7)$$

Iterační proces LA je následující:

Krok 1: Výpočet výchozích souřadnic lokalizovaného objektu X^0 a Y^0 v podobě vážených průměrů:

$$X^0 = \frac{\sum_{j=1}^n q_j X_j}{\sum_{j=1}^n q_j} \quad (7.8)$$

$$Y^0 = \frac{\sum_{j=1}^n q_j Y_j}{\sum_{j=1}^n q_j} \quad (7.9)$$

Krok 2: Výpočet první iterace souřadnic X_I a Y_I objektu pomocí rovnic 7.6 a 7.7 včetně výpočtu hodnoty účelové funkce (Z_I).

Krok 3: Porovnání hodnoty účelové funkce pro dvě poslední iterace. Je-li ΔZ blízké nule, je možné ukončit proces iterace.

Komplikovanější je proces lokalizace více objektů. V tomto případě je nutné kombinovat LA s heuristikami pro určení přiřazení zákazníka objektu. Každý zákazník je v tomto případě přiřazen nejbližšímu objektu, což zajistí minimální dopravní náklady. Umístění objektů se v tomto případě liší způsobem přiřazování zákazníků, kdy zákazníci nejsou přiřazováni objektu jeden za druhým, ale je vždy přiřazena celá skupina zákazníků najednou. Celý proces lokalizace může být popsán následovně:

Krok 1: Výchozí lokace objektů (m) jsou náhodně generovány uvnitř oblasti, ve které se nacházejí zákazníci.

Krok 2: Každý zákazník je přiřazen nejbližšímu lokalizovanému objektu. Dále je aplikován LA a je získána optimální lokace objektů vůči relevantním zákazníkům.

Krok 3: Opakováním Kroku 1 a Kroku 2 získáváme nová přiřazení objektů zákazníkům s tím rozdílem, že poloha objektů je generována pomocí algoritmu simulovaného žíhání, tj. heuristiky používané v úlohách kombinatorické optimalizace.

Krok 4: Po každé iteraci je sledována hodnota účelové funkce a v případě dosažení určité hodnoty účelové funkce je výpočet zastaven.

Nevýhodou tohoto postupu je nutnost generovat náhodně počáteční řešení, což velmi ovlivňuje efektivitu a rychlost algoritmu. Výstupem Fáze 1 je optimální lokace skladů, ze kterých jsou obsluhováni zákazníci. Tato lokace má podobu rovinných souřadnic. Dalším výstupem je matice vzdáleností, která obsahuje informace o vzdálenosti mezi všemi

zákazníky a všemi lokalizovanými objekty navzájem. Matice vzdáleností se používá při stanovování rozvozových tras. Na základě rovinných souřadnic je možné stanovit GPS souřadnice lokalizovaných objektů, a tím zasadit tyto objekty do reálných mapových podkladů. Poloha lokalizovaných objektů musí být dále korigována na základě dostupnosti potřebné infrastruktury (dopravní cesty apod.) či geografických podmínek.

Výstupy LA

- rovinné a GPS souřadnice lokalizovaných skladů,
- teritoria (přiřazení sklad – prodejny),
- matice vzdáleností.

Fáze 2: Návrh systému řízení zásob ve skladech pomocí PSMS

Cílem této fáze je zvolit optimální systém řízení zásob ve skladech a stanovit potřebnou výši disponibilní skladovací kapacity. K tomu je využita PSMS. Jedná se o programovou proceduru, která se zaměřuje na minimalizaci průměrné zásoby držené ve skladech při respektování požadavků prodejen na výši a rychlost dodávek zboží a s ohledem na omezení vyplývající z charakteristik výroben. PSMS je vytvořena v prostředí MS Excel.

Vstupní data do PSMS

- počet umístěných skladů (z Fáze 1),
- teritoria (z Fáze 1),
- požadavek skladů na dodaná množství produktů v určitém čase,
- termín vyřízení objednávky požadovaný sklady a prodejny,
- dopravní kapacity a rychlost dopravy mezi výrobnami a sklady,
- požadovaná úroveň služeb ve formě ukazatele fill rate,
- charakteristiky výroben jako kapacita, optimální a minimální výše výrobní dávky, průběžná doba výroby apod.

Princip fungování PSMS

PSMS je založeno na rekapitulaci pohybů skladové zásoby držené sklady a řízené pomocí vybraného systému řízení zásob. Nechť pro $j = 1, 2, \dots, n$ zákazníků $w_{j,i} = 1$, když je tento obsluhován jedním z $i = 1, 2, \dots, m$ skladů dle přiřazení získaného ve fázi 1.

V opačném případě necht' $w_{j,i} = 0$. Necht' poptávka po $k = 1, 2, \dots, o$ položkách poptávaných zákazníky ($Demand_{k,t,j}$) je k dispozici pro $t = 1, 2, \dots, T$ období. Necht' $l = 1, 2, \dots, q$ výrobem vyrábí poptávané položky. Vstupy do PSMS jsou reprezentovány termínem vyřízení objednávky výrobem ($LeadTime_{k,l}$), počáteční zásobou skladovaných položek ($StartingStock_{k,i}$) a agregovanou poptávkou po k -té položce ($Demand_{k,t,i}$), vyjádřenou pro $t = 1, 2, \dots, T$ období jako:

$$Demand_{k,t,i} = \sum_{j=1}^n Demand_{k,t,j} \cdot w_{j,i} \quad (7.10)$$

V každém období t jsou pohyby skladové zásoby reprezentovány splněním požadavků prodejen (snížení zásoby) a příchozem doplňujících objednávek z výrobem (zvýšení zásoby). Necht' simulace začíná v období $t = 1$ a necht' počáteční stav období ($IS_{k,t,i}$) je reprezentován počáteční zásobou ($StartingStock_{k,i}$). Nejprve je nastavena hladina aktuální zásoby ($CurrentStock_{k,i}$), která je rovna počátečnímu stavu v daném období (tedy pro $t = 1 \rightarrow CurrentStock_{k,i} = IS_{k,t,i} = StartingStock_{k,i}$). Dále je aktuální výše zásoby navýšena o příchozí objednávku ($AO_{k,t,i}$), pokud nějaká existuje (tedy pro $t = 1 \rightarrow CurrentStock_{k,i} = CurrentStock_{k,i} + AO_{k,t,i}$). Protože v kanále může být více než jedna dodávka, což je možné v případě, že $LeadTime_{k,l}$ je delší než čas mezi dvěma objednávkami, celkové objednané množství ($TOA_{k,i}$) musí být sníženo o objednané množství ihned po svém příchodu (tedy pro $t = 1 \rightarrow TOA_{k,i} = TOA_{k,i} - AO_{k,t,i}$). Následně je aktuální skladová zásoba snížena o poptávku (tedy pro $t = 1 \rightarrow CurrentStock_{k,i} = CurrentStock_{k,i} - Demand_{k,1,i}$). V případě nedostatečné aktuální skladové zásoby je poptávka uskutečněna pouze částečně. Chybějící množství ($MQ_{k,t,i}$) je zaznamenáno jako rozdíl mezi poptávkou a aktuální zásobou (tedy pro $t = 1 \rightarrow MQ_{k,t,i} = Demand_{k,1,i} - CurrentStock_{k,i}$) a aktuální zásoba je vynulována ($CurrentStock_{k,i} = 0$). Simulace nebere v úvahu doobjednávání, což znamená, že v případě, že poptávka prodejny v období t je větší než aktuální zásoba skladu, dochází ke ztrátě prodeje. V dalším kroku kontroluje simulace, zda je nutné zadat objednávku a její velikost. Objedávka je vytvořena, jestliže aktuální zásoba navýšená o celkové objednané množství je menší nebo rovna signálnímu stavu zásob (r). Příchod objednávky zadané v období t bude učiněn na počátku období $t + LeadTime_{k,l} + 1$. Signální hladina závisí na vybraném typu řízení stavu zásob. Simulace pracuje se dvěma základními typy řízení stavu zásob. V případě nasazení Q-systému je objednávací množství konstantní a množství doručené v období $t + LeadTime_{k,l} + 1$ (tedy pro $t = 1 \rightarrow AO_{1 + LeadTime_{k,l} + 1}$) se rovná objednávacímu množství (Q). V případě nasazení PQ-systému je množství objednané v období $t + LeadTime_{k,l} + 1$ (tedy pro $t = 1 \rightarrow AO_{1 + LeadTime_{k,l} + 1}$) do výše horní objednávací meze (x_h) sníženo o aktuální zásobu a celkové objednané množství. Po vytvoření objednávky je objednané množství navýšeno o vytvořenou objednávku (tedy pro $t = 1 \rightarrow TOA_{k,i} = TOA_{k,i} + AO_{1 + LeadTime_{k,l} + 1}$) a konečný stav období ($FS_{k,t,i}$) je nastaven na hodnotu aktuální zásoby ($FS_{k,1,i} = CurrentStock_{k,i}$). Dále simulace pokračuje s pohybem zásob v období $t = 2, 3, \dots, T$. Všechna tato období začínají z počátečního stavu ($IS_{k,t>1,i}$), který je roven konečnému

stavu předcházejícího období ($FS_{k,t-1,i}$). Výhodou této simulační struktury je možnost stanovit efektivnost skladování a objednávání a dále také schopnost uspokojit poptávku přímo ze skladové zásoby. Ke stanovení efektivnosti skladování a objednávání je vypočítána průměrná zásoba z konečných stavů zásob všech období jako:

$$x_{avg,k,i} = \frac{\sum_{t=1}^T FS_{k,t,i}}{T} \quad (7.11)$$

Schopnost uspokojit poptávku přímo ze skladové zásoby je vypočítána ve formě fill rate ($FL_{k,i}$). Fill rate reprezentuje poptávku, která může být uspokojena z aktuální zásoby. K výpočtu fill rate pro skladovanou položku jsou použita chybějící množství získaná během doby běhu simulačního modelu vedoucí k:

$$FL_{k,i} = 1 - \frac{\sum_{t=1}^T MQ_{k,t,i}}{\sum_{t=1}^T Demand_{k,t,i}} \quad (7.12)$$

Pomocí ukazatele fill rate je možné nastavit různou úroveň služeb pro každou položku dle jejího významu například z pohledu výše generovaných tržeb. Aby bylo možné dosáhnout požadované úrovně služeb ve formě fill rate, je třeba řídit PSMS pomocí vhodné kombinace řídicích parametrů, které charakterizují vybraný systém řízení zásob. Jinými slovy, když bude například požadovaná fill rate pro nějakou položku 98 %, celková poptávka v období T 100 kusů a vybraný systém řízení zásob bude Q-systém, správná kombinace signálního stavu (r) a velikost doplňující objednávky (Q) musí zajistit, že celkové chybějící množství v období T nepřekročí 2 kusy. Totéž platí pro PQ-systém a kombinaci parametrů signální stav (r) a horní objednávací mez (x_h). Pro vybraný systém řízení zásob nicméně existuje velké množství kombinací řídicích parametrů, které zajistí splnění ukazatele fill rate pro konkrétní skladovanou položku. Nabízí se tedy otázka, která z kombinací řídicích parametrů je ta nejvhodnější. Je to ta kombinace, která zajistí nejnížší průměrnou zásobu. K nalezení této kombinace je nutné provést totální enumeraci obou řídicích parametrů a spustit PSMS pro všechny takto vytvořené kombinace. Totální enumerace parametru signální stav znamená generovat hodnoty z intervalu:

$$\langle 0; \sum_{t=1}^T Demand_{k,t,i} \rangle \quad (7.13)$$

respektive pro velikost doplňující objednávky či horní objednávací mez hodnoty z intervalu:

$$\langle 1; \sum_{t=1}^T Demand_{k,t,i} \rangle \quad (7.14)$$

PSMS poté běží opakovaně pro každou kombinaci řídicích parametrů a je vybrána kombinace s nejnížší hodnotou průměrné zásoby, která zajistí splnění požadované fill rate. Na základě umístěných doplňujících objednávek a při respektování dodacích termínů výroben jsou naplánovány dodávky zboží z výroben do skladů. Podmínkou je co nejvyšší

využití kapacity dopravců, kteří tyto dodávky zajišťují. Tok zboží přicházející do skladů zvýšený o počáteční zásobu a snížený o tok odcházejícího zboží slouží jako informace o maximální potřebné skladovací kapacitě.

Výstupy PSMS

- potřeba skladovací kapacity ve skladech,
- plán dodávek zboží z výroby do skladů.

Fáze 3: Simulace materiálových toků s využitím CWS pro plánování rozvozových tras

Cílem této fáze je navrhnout vnitřní layouty skladů a zjistit optimální počet pracovníků a manipulační techniky pro realizaci materiálových toků. K tomuto účelu je v prostředí softwaru Witness provedena simulace materiálových toků. Součástí této simulace je také konstrukce efektivních rozvozových tras mezi sklady a obsluhovanými prodejny tak, aby bylo dosaženo maximální úspory ujeté vzdálenosti při respektování požadavků prodejen na množství dodaného zboží a požadovaný termín vyřízení objednávky. Ke konstrukci rozvozových tras je využit CWS.

Vstupní data do CWS

- počet lokalizovaných skladů (z Fáze 1),
- teritoria (z Fáze 1),
- matice vzdáleností (z Fáze 1),
- požadavky prodejen na dodaná množství produktů v určitém čase,
- termín vyřízení objednávky požadovaný prodejny,
- maximální dopravní kapacity a rychlost dopravy při obsluze navržených rozvozových tras.

Princip fungování CWS

CWS je jedním z nejčastěji citovaných algoritmů používaných k řešení problému obchodního cestujícího, tj. vehicle routing problem (VRP). Základní strukturu lze najít v Clarke and Wright (1964), modifikace a využití jsou popsány např. v Holmes and Parker (1976), Dror and Trudeau (1986) či Paessens (1988). V simulaci materiálových toků je CWS využito k řešení kapacitního problému obchodního cestujícího, tj. capacitated vehicle routing problem (CVRP). V CVRP jsou požadavky zákazníků v podobě poptávaných množství produktů obsluhovány dopravci, kteří dopravují zboží ze skladů. Nechť každý

dopravce disponuje stejnou kapacitou a každý zákazník poptává určité množství produktu. Dále je k dispozici matice vzdáleností se vzdálenostmi mezi všemi zúčastněnými sklady a zákazníky. Základní myšlenka CWS je velmi jednoduchá. Necht' nějaký i -tý sklad reprezentuje depo D_i , zákazníci reprezentují $j = 1, 2, \dots, n$ výchozích míst poptávky, $j' = 1, 2, \dots, n$ konečných míst poptávky, $d(D_i, j)$ reprezentuje vzdálenost mezi nějakým bodem poptávky a depem a $d(j, j')$ reprezentuje vzdálenost mezi dvěma různými body poptávky ($j \neq j'$). Předpokládejme, že řešení VRP bude zajišťovat n dopravců a každý bude obsluhovat právě jeden z n bodů poptávky. Celková délka cesty TL tohoto řešení je:

$$TL = 2 \sum_{j=1}^n d(D_i, j) \quad (7.15)$$

Jestliže jeden dopravce obslouží dva různé body poptávky (j, j') během jedné cesty, celková délka cesty je zkrácena o úsporu:

$$s(j, j') = d(D_i, j) + d(D_i, j') - d(j, j') \quad (7.16)$$

Čím větší je $s(j, j')$, tím více je žádoucí kombinovat j a j' v jedné cestě. CWS může být nyní definováno následujícími kroky:

Krok 1: Spočti úspory $s(j, j')$ pro každou dvojici (j, j') bodů poptávky pomocí rovnice 7.16.

Krok 2: Seřaď úspory $s(j, j')$ sestupně dle jejich výše do podoby seznamu. Zpracuj seznam úspor od prvního záznamu s největší úsporou $s(j, j')$.

Krok 3: Pro úspory $s(j, j')$, zahrň úsek (j, j') do okruhu v případě, že kapacita dopravního prostředku je dostatečná a čas strávený na cestě nepřekročí maximální povolenou dobu danou termínem vyřízení objednávky a jestliže:

- a: Bud' ani j , ani j' nebylo ještě přiřazeno na okruh, v tomto případě je založen nový okruh s oběma body j a j' .
- b: Nebo právě jeden z těchto dvou bodů (j nebo j') je již přiřazen na trasu a tento bod není vnitřním bodem této trasy. V tomto případě je úsek (j, j') přidán do stejného okruhu.
- c: Nebo oba body j a j' již byly přiřazeny do dvou existujících okruhů a žádný z těchto bodů není vnitřní, v tomto případě jsou tyto cesty spojeny.

Krok 4: Jestliže seznam úspor $s(j, j')$ nebyl ještě vyčerpán, vrať se ke kroku 3 a zpracuj další záznam v seznamu; v opačném případě *Stop*: řešení VRP sestává z okruhů vytvořených v kroku 3. Všechny body, které ještě nebyly přiřazeny k okruhu v kroku 3,

musí být obslouženy okruhem, který začíná v depu D_i , navštíví nepřirazené body a vrátí se zpět.

Výstupy CWS

- Plán rozvozů.

Výstupy CWS, tj. plán rozvozů, a výstupy PSMS, tj. plán dodávek zboží z výroby do skladů, slouží jako vstupy do obecného simulačního modelu materiálových toků. Tato simulace je využita k ověření návrhu layoutů skladů a zjištění optimálního počtu zdrojů, tj. pracovníků a manipulační techniky pro realizaci materiálových toků. Simulace pracuje s materiálovým tokem rozloženým do pohybů: doprava zboží z výroby do skladu, vykládka zboží v příjmové zóně skladu, kontrola množství a označení zboží štítkem, zaskladnění zboží do regálu, kompletace objednávek, balení, nakládka zboží, rozvoz zboží do prodejen. Realizace pohybů doprava zboží z výroby do skladu a rozvoz zboží do prodejen probíhá v režii externího dopravce, a proto není v simulaci uvažováno přiřazení zdrojů těmto pohybům. Pohyby tvořící materiálový tok jsou simulovány pomocí ZMS na základě požadavků na provedení pohybu generovaných generátory požadavků, které pracují se vstupy získanými CWS a PSMS. Vykládku zboží v příjmové zóně skladu a nakládku zboží zajišťují zdroje pracovníci a ručně vedený vozík bez zdvihu. Uvažovaná rychlost chůze pracovníka je 3 km/h, trvání manipulace u vykládky a nakládky je uvažováno 0,5 min/pal. Kontrolu množství a označení zboží štítkem provádí zdroj pracovníci a trvá 0,3 min/pal. Zaskladnění zboží do regálu je chaotické a zajišťují ho zdroje pracovníci a elektrický ručně vedený vozík se zdvihem, přičemž v simulaci je uvažována rychlost pohybu pracovníka 5 km/h a trvání manipulace u zakládání zboží do regálu je 0,75 min/pal. Stejně zdroje pohybující se stejnou rychlostí jsou využívány při kompletaci zboží. Kompletace je iniciována na základě plánu rozvozů vždy tak, aby se zboží dostalo včas do prodejen. Trvání manipulace u kompletace je 1,1 min/1 kompletovaná položka. Při kompletaci je dodržováno pravidlo first in first out a pracovníci se v regálových uličkách pohybuje vždy od nejvzdálenější lokace vzhledem k čelu regálu k lokaci nejbližší. Balení probíhá na baličce, kterou obsluhuje zdroj pracovníci a trvá 1,6 min/pal. Vstupní data do simulace pokrývají období jednoho roku.

7.1.3 Posouzení efektivity navržené struktury distribučního systému – analýza nákladů na distribuci

Ke stanovení efektivity navržené struktury distribučního systému je provedena analýza nákladů na distribuci. Vstupní data této analýzy pocházejí z fází 1–3. Jedná se o:

- počet lokalizovaných skladů a jejich polohu,

- rozvozové trasy spojující sklady a prodejny charakterizované ujetou vzdáleností a dopraveným množstvím,
- dodávky zboží z výroben do skladů charakterizované ujetou vzdáleností a dopraveným množstvím,
- průměrnou zásobu skladovaných položek,
- potřebu skladovací kapacity ve skladech,
- počty pracovníků a manipulační techniky potřebné k realizaci materiálových toků ve skladech.

Na základě $q = 1, 2, \dots, u$ rozvozových tras spojujících sklady a prodejny, které jsou charakterizovány ujetou vzdáleností a dopravovaným množstvím, lze spočítat dopravní náklady C_1 jako:

$$C_1 = \sum_{q=1}^u RD_q r_t \quad (7.17)$$

kde RD_q je ujetá vzdálenost (např. v km) a r_t je sazba účtovaná dopravě závislá na kapacitě použitého vozidla.

Podobně na základě $x = 1, 2, \dots, y$ dodávek zboží z výroben do skladů lze vyjádřit dopravní náklady C_2 jako:

$$C_2 = \sum_{x=1}^y RD_x r_t \quad (7.18)$$

kde RD_x je ujetá vzdálenost (opět např. v km).

Provozní náklady skladů C_3 lze vyjádřit jako:

$$C_3 = PP \cdot r_p + PV \cdot r_v \cdot \sum_{i=1}^m c_p K_i \quad (7.19)$$

kde PP je počet pracovníků ve skladech, r_p roční osobní náklady na 1 pracovníka skladu, PV je počet elektrických ručně vedených vozíků se zdvihem používaných k manipulaci ve skladech, r_v roční náklady spojené s pronájmem těchto vozíků, c_p nepřímé náklady na jedno paletové místo a K_i potřeba skladovací kapacity v m lokalizovaných skladech.

Průměrná výše zásoby k skladovaných položek lokalizovaná v m skladech slouží k výpočtu nákladů vyplývajících z vázanosti kapitálu v zásobách C_4 :

$$C_4 = r_a \cdot \sum_{i=1}^m \sum_{k=1}^o x_{avg,k,i} \cdot p_k \quad (7.20)$$

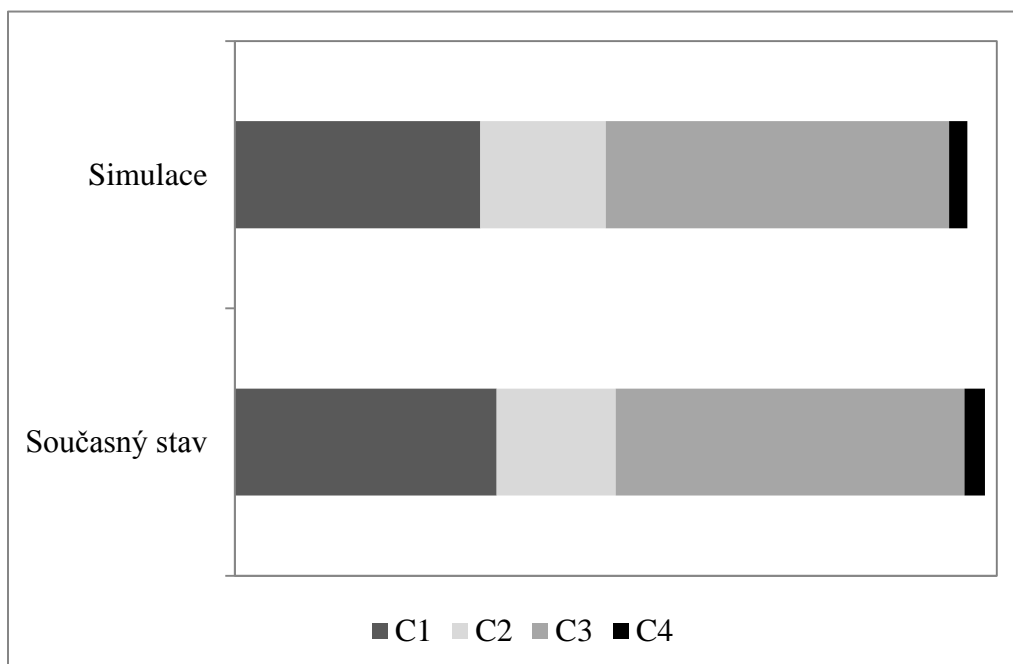
kde r_a reprezentuje diskontní sazbu a p_k je prodejní cena skladované položky.

Celkové náklady na distribuci C použité pro srovnání efektivnosti různých struktur distribučního systému jsou:

$$C = \sum_{i=1}^4 C_i \quad (7.21)$$

7.1.4 Simulace současné struktury distribučního systému

Simulace současného uspořádání distribučního systému se 6 sklady je provedena s vynecháním Fáze 1. Výstupy simulace přepočítané do podoby nákladů na distribuci $C_I - C_4$ byly porovnány se skutečnými náklady (viz Obr. 7.5).

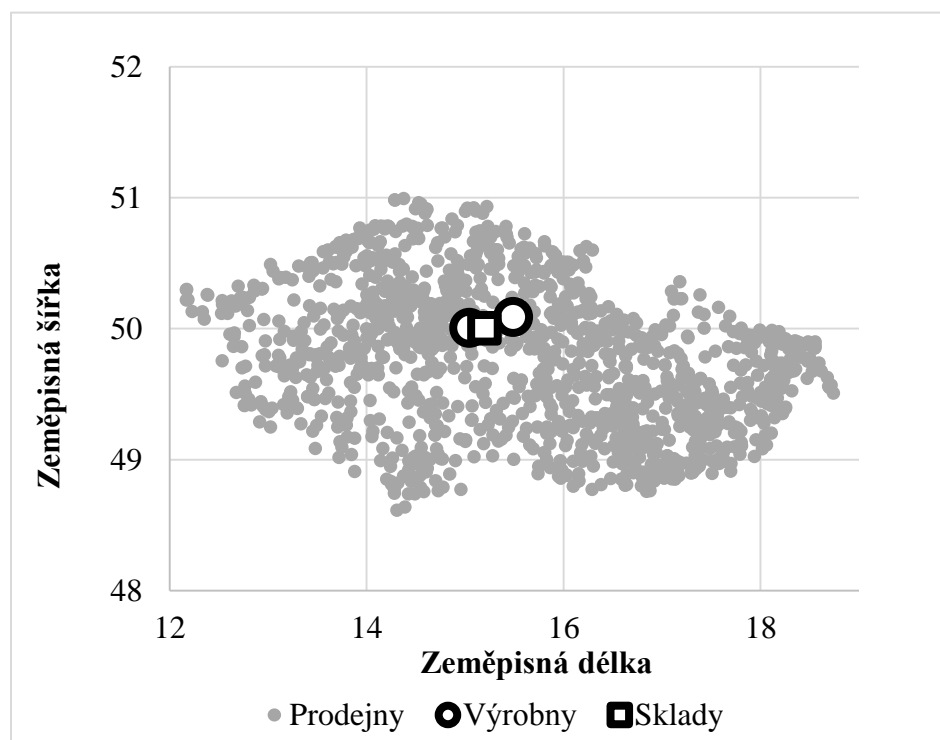


Obr. 7.5 Srovnání skutečných nákladů na distribuci současného uspořádání distribučního systému se simulací

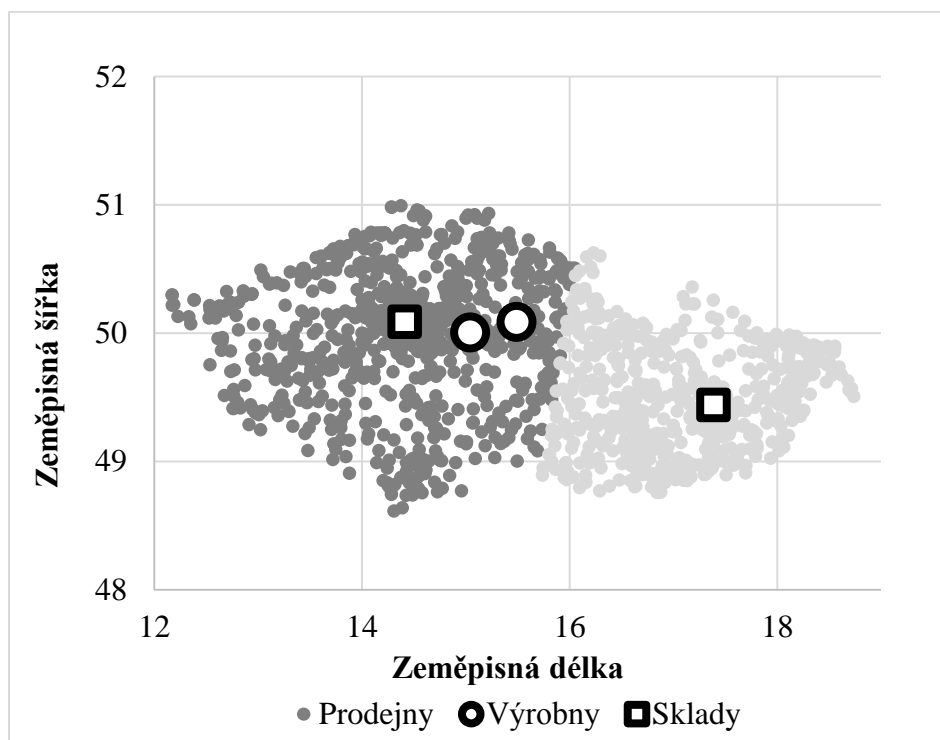
Výsledky simulace současného stavu poukázaly na rezervy v plánování rozvozových tras mezi sklady a prodejny (viz nižší dopravní náklady C_I simulace oproti současnému stavu) a na vyšší průměrný stav zásob držený ve skladech (viz nižší vázanost kapitálu v zásobách v podobě nákladů C_4 simulace oproti současnému stavu), který vede k vyšší skutečné potřebě skladovacích kapacit. Nejedná se nicméně o podstatný rozdíl co do případné absolutní výše úspory nákladů, neboť rozdíl mezi skutečnými celkovými náklady na distribuci a celkovými náklady na distribuci získanými simulací se pohybuje okolo 2 %.

7.1.5 Redesign distribučního systému

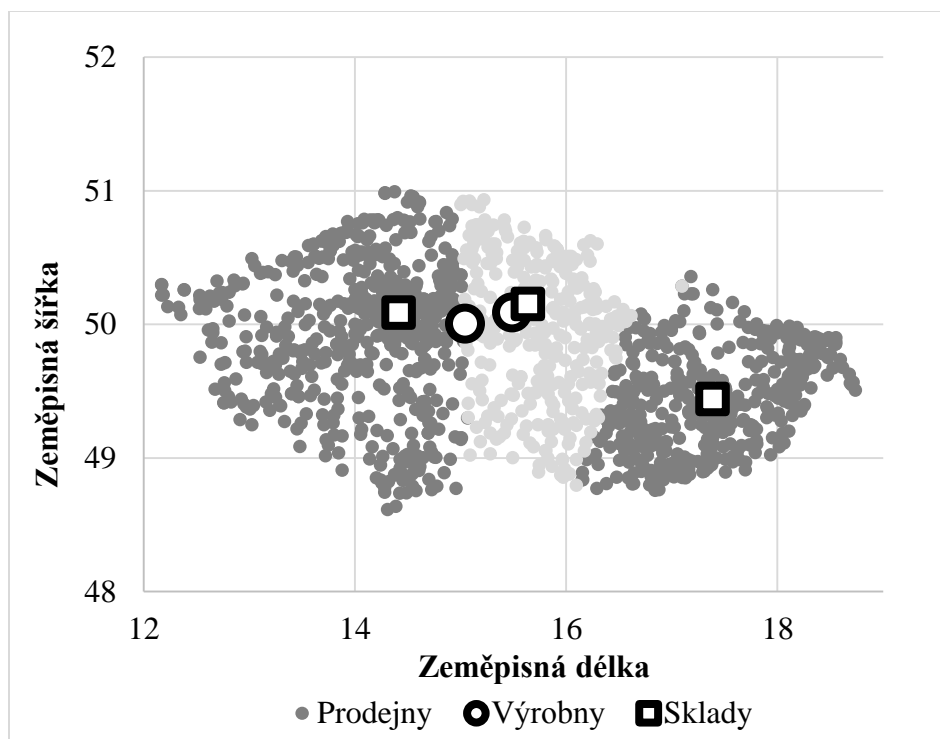
Po provedení simulace současného uspořádání distribučního systému byla provedena simulace 6 variant uspořádání alternativního, které se liší počtem lokalizovaných skladů. Tento počet se pohybuje od 1 do 6. Maximální počet skladů je zvolen s ohledem na současný stav systému a také s ohledem na termín vyřízení objednávky požadovaný prodejny, rozlohu obsluhovaného regionu a hustotu silniční sítě. Pro každou variantu byla nejprve pomocí Fáze 1 zjištěna poloha skladů, přičemž rozdělení obsluhovaných prodejen do teritorií bylo provedeno opět na základě minimální vzdálenosti mezi prodejnou a všemi sklady (viz Obr. 7.6–Obr. 7.11).



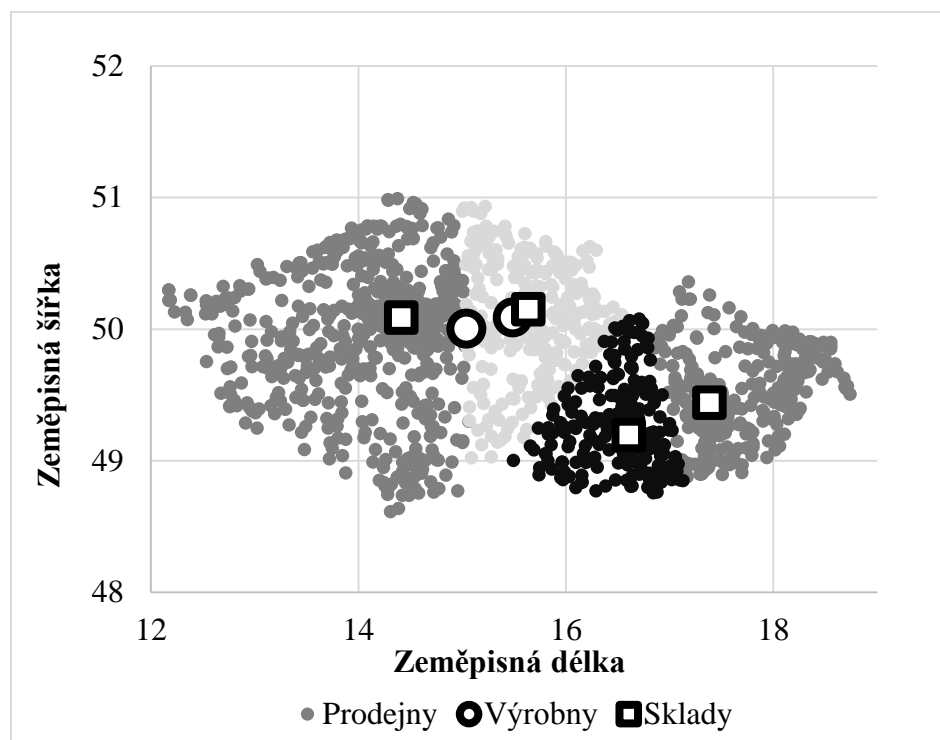
Obr. 7.6 Umístění skladu a návrh teritorií – varianta s 1 skladem



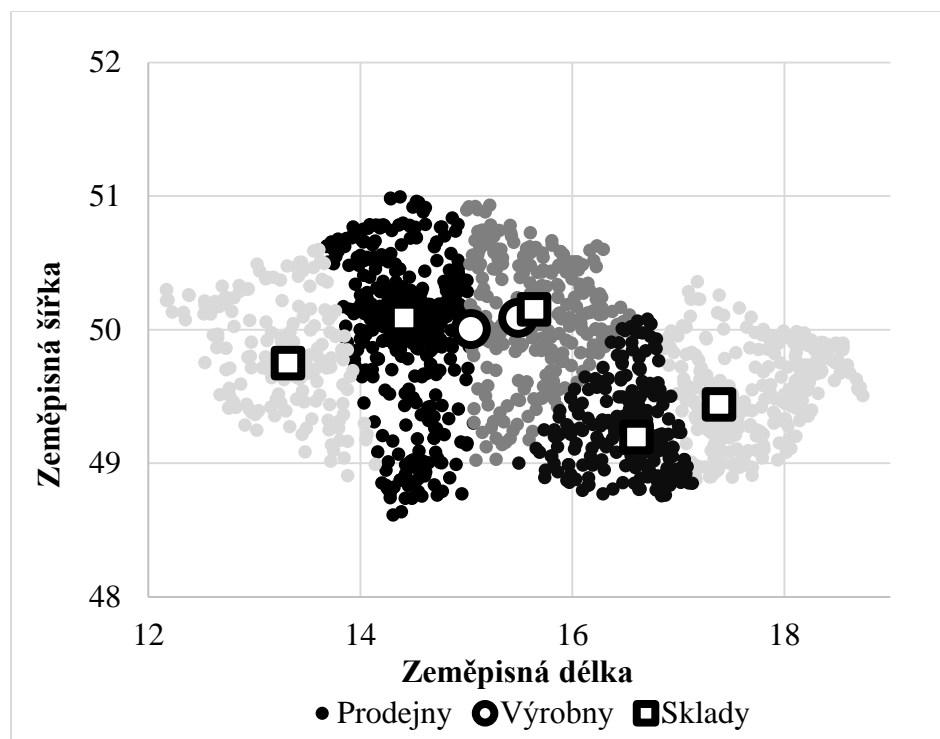
Obr. 7.7 Umístění skladů a návrh teritorií – varianta se 2 sklady



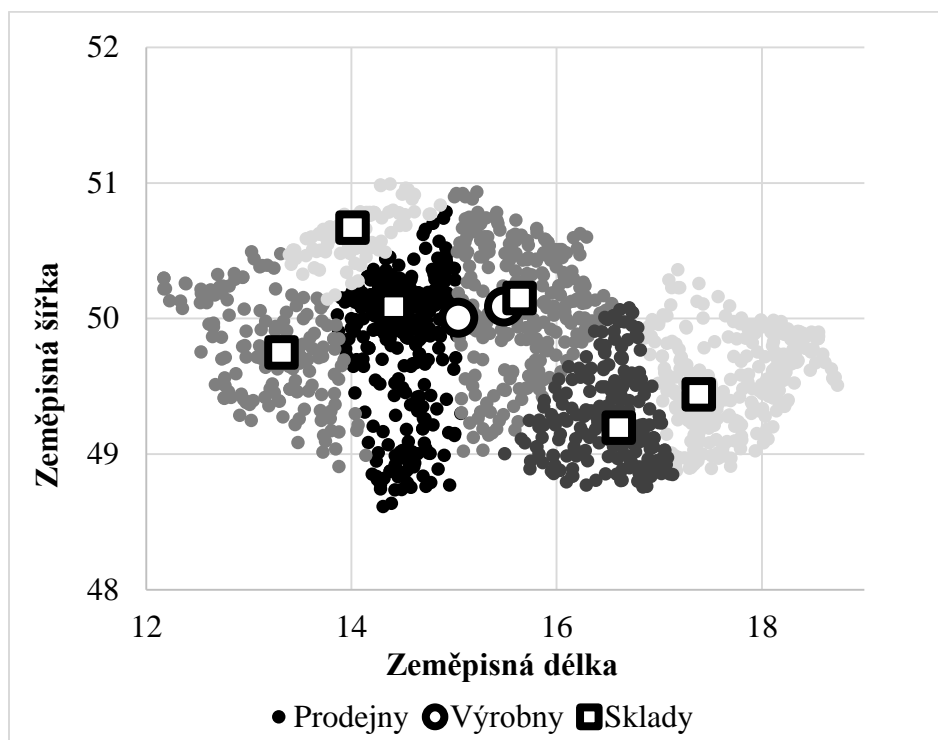
Obr. 7.8 Umístění skladů a návrh teritorií – varianta se 3 sklady



Obr. 7.9 Umístění skladů a návrh teritorií – varianta se 4 sklady

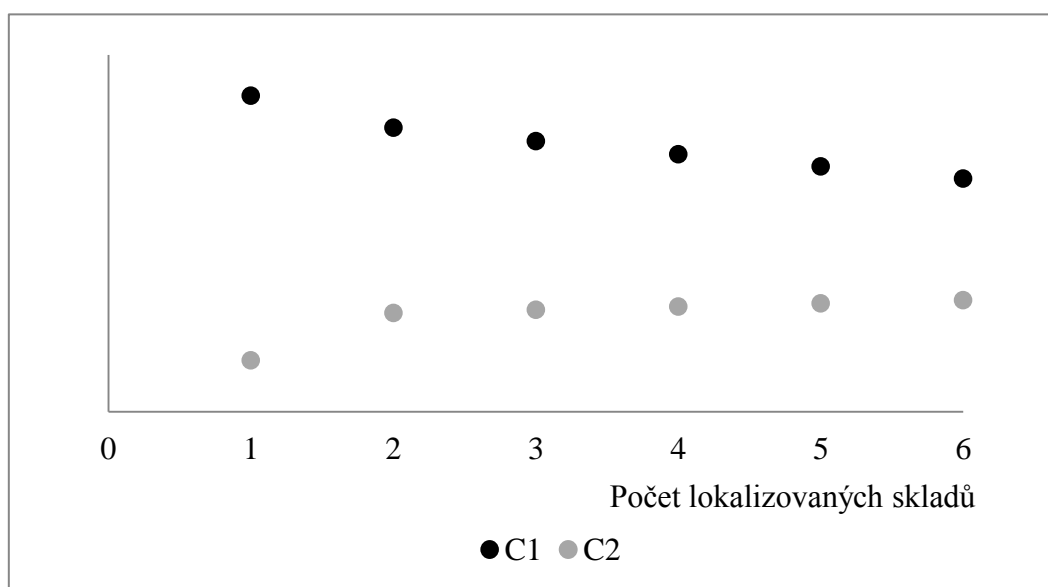


Obr. 7.10 Umístění skladů a návrh teritorií – varianta s 5 sklady



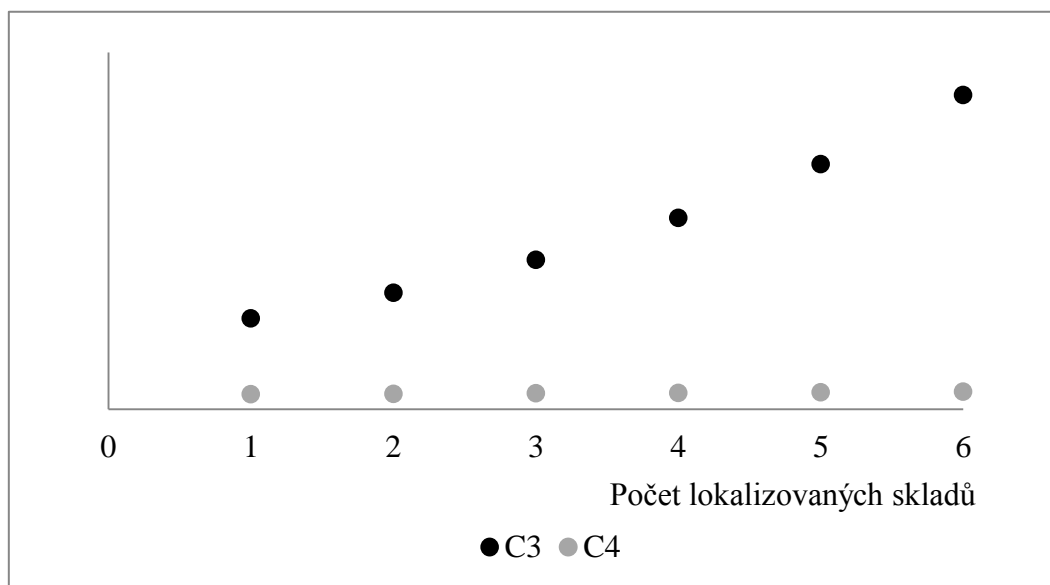
Obr. 7.11 Umístění skladů a návrh teritorií – varianta se 6 sklady

Na základě lokalizace skladů a po provedení fází 2 a 3 byly získány vstupy do analýzy nákladů na distribuci. Vývoj nákladů na dopravu zboží mezi sklady a prodejnami C_1 a mezi výrobními a sklady C_2 v závislosti na počtu skladů jsou zachyceny na Obr. 7.12.



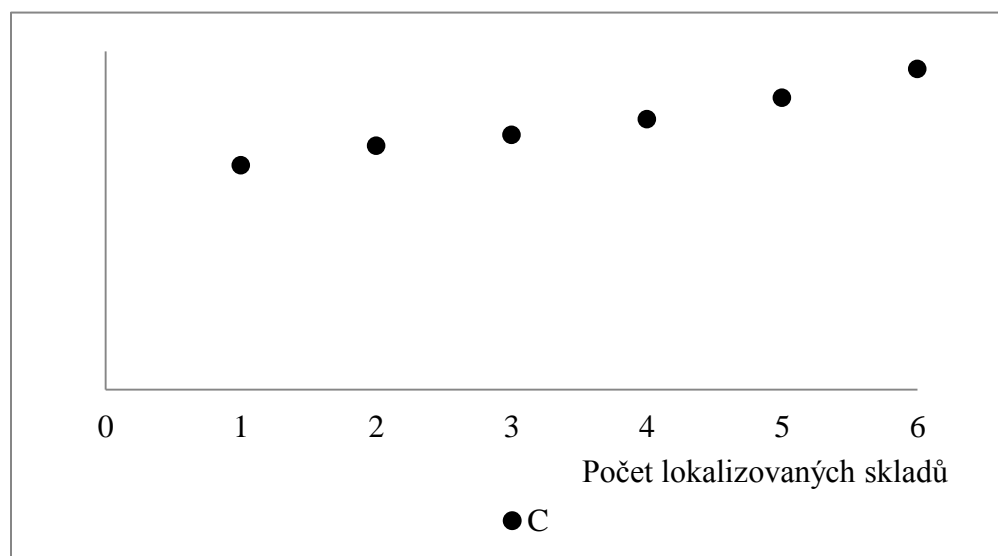
Obr. 7.12 Dopravní náklady C_1 a C_2

Vývoj provozních nákladů skladů C_3 a nákladů z vázanosti kapitálu v zásobách C_4 v závislosti na počtu skladů jsou zobrazeny na Obr. 7.13.



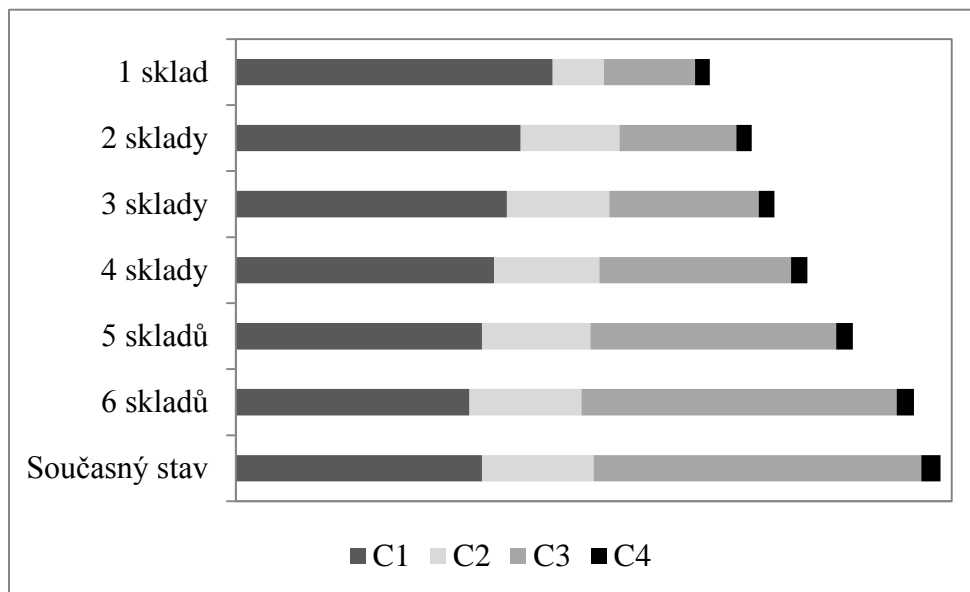
Obr. 7.13 Provozní náklady skladů C_3 a náklady z vázanosti kapitálu v zásobách C_4

Vývoj celkových nákladů na distribuci C v závislosti na počtu skladů zachycuje Obr. 7.14.



Obr. 7.14 Celkové náklady na distribuci C

Celkové náklady na distribuci všech simulovaných variant byly následně porovnány s náklady na distribuci současného stavu (viz následující obrázek) a byly vyčísleny úspory celkových nákladů na distribuci ΔC (viz následující tabulka).



Obr. 7.15 Porovnání skutečných nákladů na distribuci s náklady na distribuci získanými simulací pro 6 různých variant

Tabulka 7.1 Úspory celkových nákladů na distribuci

Varianta	1 sklad	2 sklady	3 sklady	4 sklady	5 skladů	6 skladů
ΔC	4 %	12 %	19 %	24 %	27 %	33 %

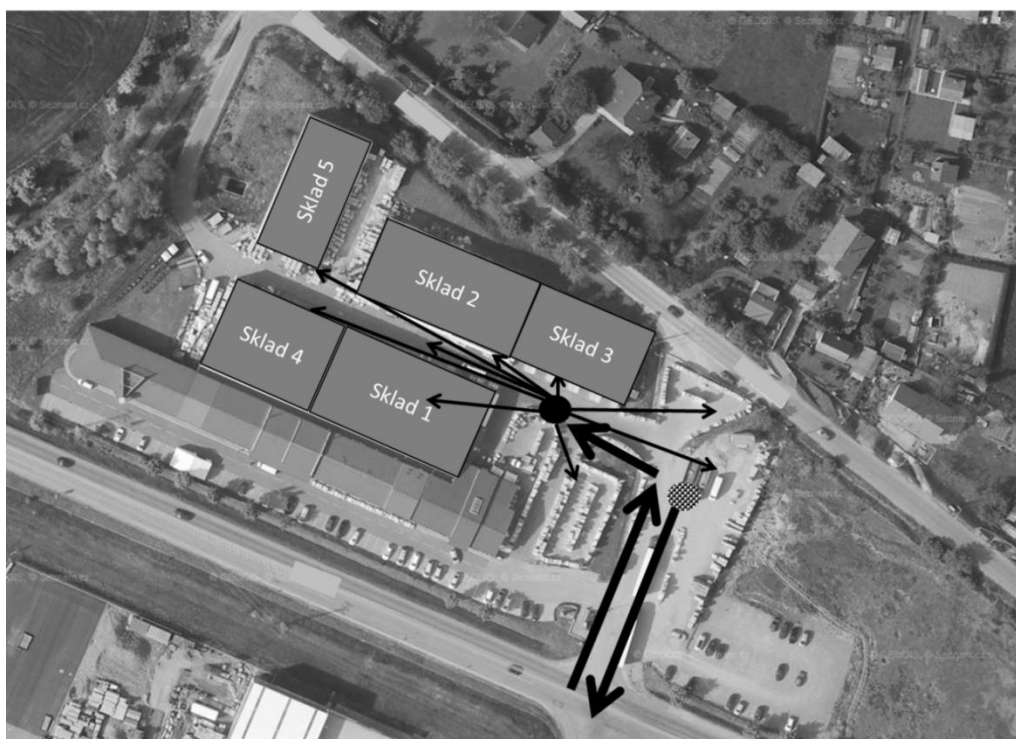
Odhadovaná úspora ročních nákladů na distribuci je cca 33 % v případě varianty s jedním skladem.

7.2 Návrh koncepce logistiky v centrálním skladu společnosti zabývající se nákupem a prodejem stavební chemie a obkladů

7.2.1 Současný stav logistiky v centrálním skladu společnosti

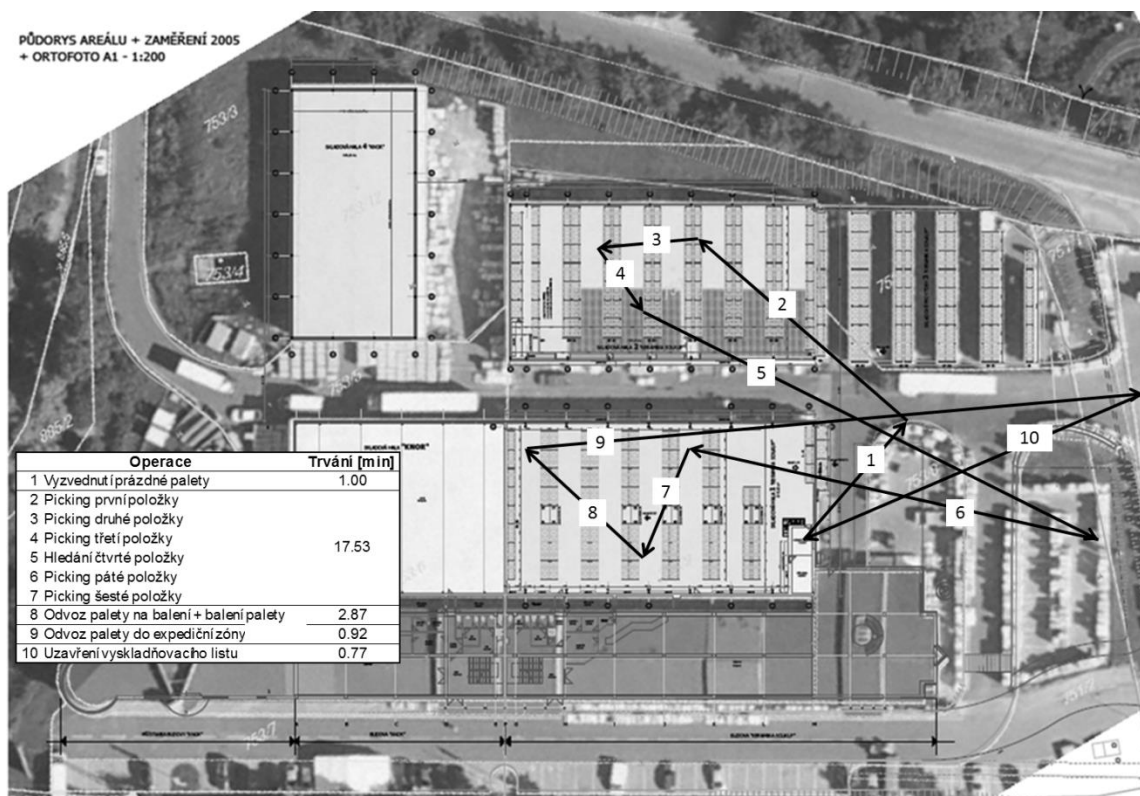
Společnost nakupuje zboží z oblasti stavební chemie a obklady za účelem dalšího prodeje na území České republiky. Jedná se o portfolio, které obsahuje zhruba 11 000 aktivních položek. Tržby společnosti v posledních 3 letech rostou řádově o desítky % za rok, což má

za následek prudký nárůst intenzity materiálových toků skladovým areálem (skladový areál společnosti viz Obr. 7.16).



Obr. 7.16 Skladový areál společnosti

Skladový areál disponuje 5 oddělenými sklady vybavenými paletovými regály či s volným skladováním na ploše s celkovou kapacitou cca 5 000 paletových míst. Sklady jsou obsluhovány čelními vysokozdviznými vozíky s pohonem na naftu. Příjem zboží probíhá v centrální části areálu (viz černý kruh na Obr. 7.16), kde vysokozdvizné vozíky vykládají palety se zbožím z nákladních automobilů, probíhá kvantitativní přejímka a palety jsou dále chaoticky zaskladněny do volných pozic ve skladech. Po fyzickém zaskladnění následuje zaskladnění systémové, kdy je do informačního systému zadáno zboží, jeho množství a lokace, na kterou bylo zboží umístěno. Při příjmu nedochází k označení zboží čárovým kódem. Vychystávání zboží probíhá na základě objednávek zákazníků přijatých prodejním oddělením, které jsou ve formě tištěných papírových výdejek předávány na sklad 1. Zde si je přebírají skladníci starající se o kompletaci objednávek. Ti objíždějí s vysokozdviznými vozíky lokace se zbožím a skládají požadovaná množství na paletu (viz Obr. 7.17).

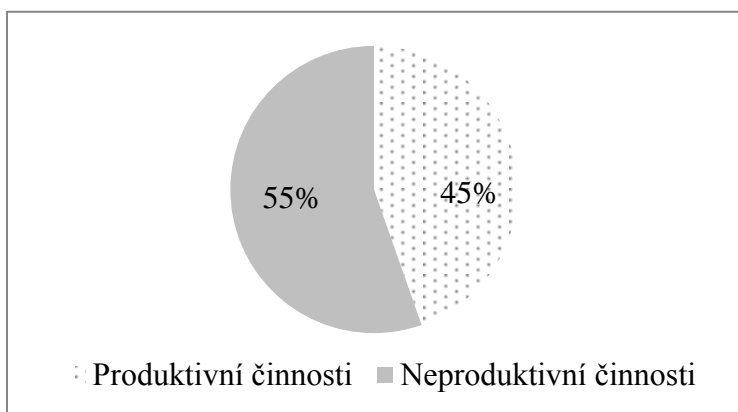


Obr. 7.17 Typický průběh fyzické kompletace objednávky zákazníka

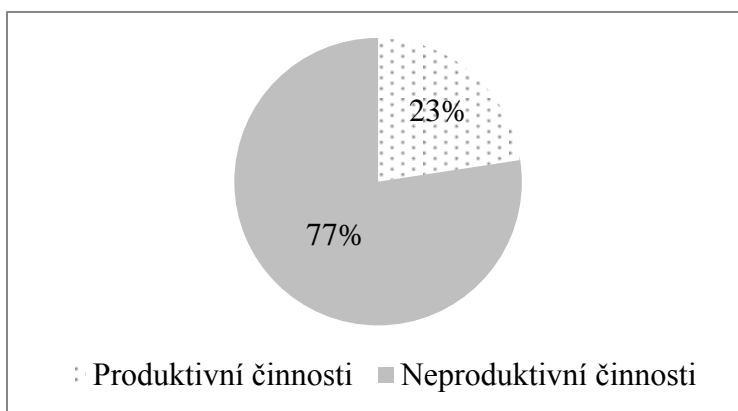
Pořadí navštívených lokací stanovuje skladník na základě zkušeností. Po kompletaci objednávky je paleta ručně zabalena do folie, převezena do prostoru expedice (viz šachovnicový kruh na Obr. 7.16) a naložena do nákladních automobilů. Stejně jako při příjmu, provádějí skladníci po kompletaci objednávky systémové vyskladnění, kdy je na příslušných lokacích v informačním systému zaznamenán množstevní úbytek zboží. Rostoucí objemy zboží protékajícího skladovým areálem a tlak na zkracování dodacích termínů řeší společnost nabíráním dalších skladníků a pronájmem manipulační techniky. To při stagnující produktivitě práce a neefektivitě nákupu vede ke komplikované dopravní situaci v areálu, hromadění skladových zásob mimo určené skladové pozice a prodlužování termínů vyřízení objednávky.

Pomocí auditu strategického a operativního nákupu společnosti bylo zjištěno, že průměrná zásoba rychloobrátkového zboží s dobou obratu menší než 30 dnů, které se na tržbách podílí z 80 %, zabírá 20 % skladové kapacity areálu, zatímco průměrná zásoba maloobrátkového zboží s dobou obratu větší než 90 dnů, které se na tržbách podílí z 5 %, zabírá 52 % skladové kapacity areálu. To potvrzuje skutečnost, že společnost nedisponuje strategií nákupu založenou na hladinovém řízení zásob a efektivní předpovědi poptávky. Z procesního auditu logistiky bylo na základě snímků pracovního času a analýzy

skutečných materiálových toků za období jednoho roku stanoveno využití zdrojů spojených s příjmem, vychystáváním a expedicí zboží (viz Obr. 7.18 a Obr. 7.19):



Obr. 7.18 Využití skladníků společnosti

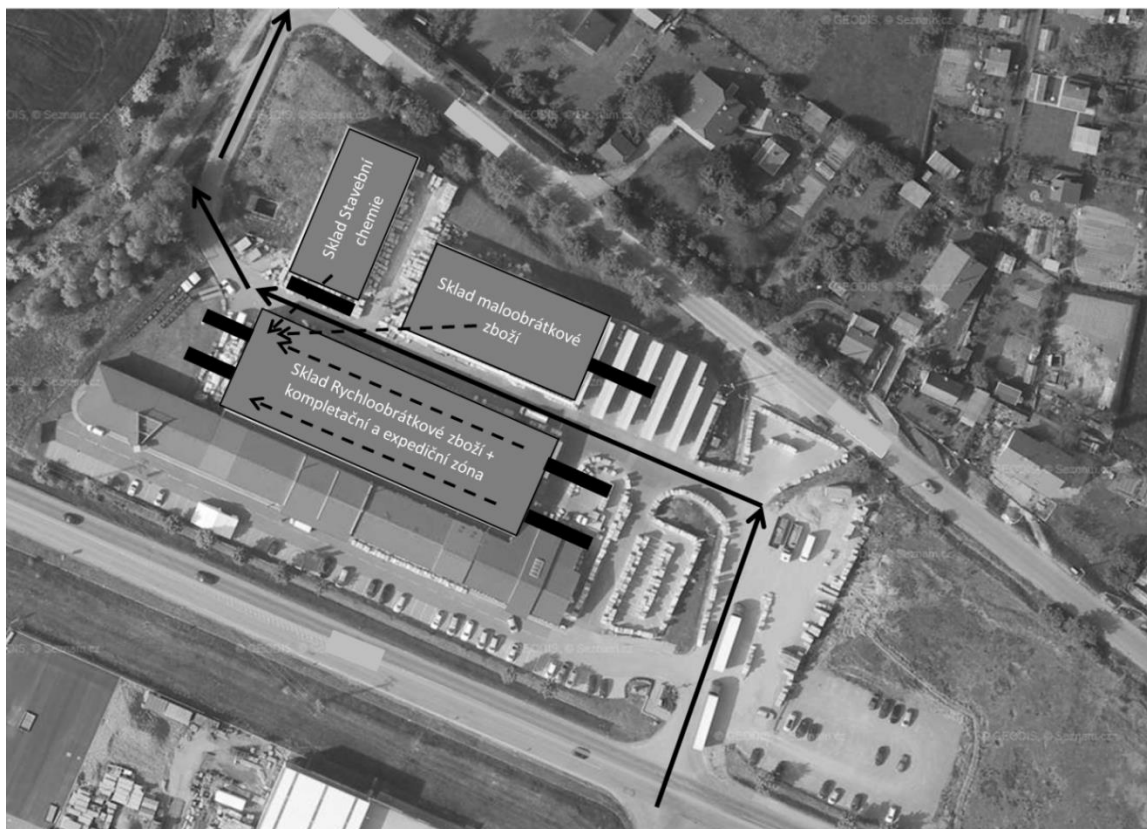


Obr. 7.19 Využití manipulační techniky používané skladníky

Mezi neproduktivní činnosti patří zejména hledání zboží v prostorách skladového areálu, nutnost přeskladňovat zboží z důvodu nedostupnosti požadované položky, čekání na uvolnění uliček v regálovém systému při fyzické kompletaci objednávek či nečinnost. Z podílu neproduktivních činností byla dále pomocí osobních nákladů a nákladů na provoz flotily manipulační techniky vyčíslena neefektivita procesů v logistice ve výši jednotek milionů Kč za rok.

7.2.2 Návrh koncepce logistiky v centrálním skladu společnosti a její modelování pomocí obecného simulačního modelu materiálových toků

Návrh koncepce logistiky v centrálním skladu společnosti se opírá o strategii nákupu založenou na hladinovém řízení zásob a efektivní předpovědi poptávky a také o koncept řízeného skladování. Pro účely řízeného skladování jsou navrženy úpravy layoutu skladového areálu, které spočívají v rozdělení skladovacích kapacit na sklady určené pro skupinu zboží obklady a skupinu zboží stavební chemie (viz Obr. 7.20).



Obr. 7.20 Úpravy layoutu skladového areálu pro koncept řízeného skladování

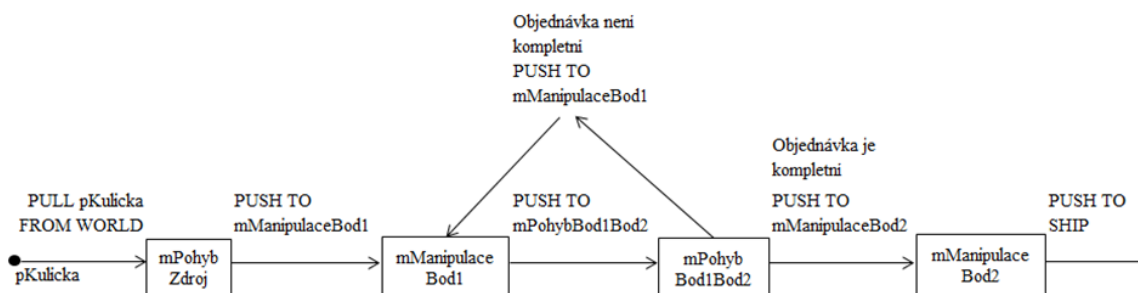
Položky ze skupiny obklady jsou dále alokovány na sklady rychloobrátkového a maloobrátkového zboží na základě historické doby obratu. Příjem do skladů skupiny obklady je uvažován z čela nákladních automobilů pomocí vykládacích ramp (celkem 3 rampy), pro sklad stavební chemie vykládkou z boku. Vykládka je realizovaná pomocí čelních vysoko zdvižných vozíků s elektrickým a naftovým pohonem. Všechny sklady jsou vybaveny regálovým systémem pro skladování palet, přičemž sklad rychloobrátkového zboží využívá technologie very narrow aisle (VNA) pro maximální využití prostoru. Z toho vyplývá, že ve skladu rychloobrátkového zboží jsou k zaskladnění palet do regálů při příjmu využívány zakladače vhodné k použití ve VNA, zatímco ve zbývajících skladech jsou k zaskladnění palet v širších uličkách využívány čelní vysoko zdvižné vozíky. Lokace pro zaskladnění jsou generovány pomocí warehouse management systému (WMS) na

základě historické obrátkovosti položek a jednoduchého pravidla „čím větší obrátkovost, tím blíže expediční zóně ve skladu rychloobrátkového zboží“. Systémové zaskladnění probíhá pomocí technologie čárových kódů, kdy při přejímce je zboží nejprve označováno papírovým štítkem a v okamžiku, kdy je zboží fyzicky umístěno na lokaci a spárováno čtečkou s označením této lokace, jsou pomocí bezdrátového přenosu dat aktualizovány informace o skladovém hospodářství v informačním systému. Kompletace objednávek zákazníků probíhá kombinovaně: ve skladech maloobrátkového zboží a stavební chemie je využíván statický systém a ve skladu rychloobrátkového zboží systém dynamický. Statický systém znamená, že skladníci se při kompletaci pohybují za zbožím, které vybírají v požadovaném množství z lokací, v dynamickém systému dopravují zakladače palety se zbožím do kompletační zóny a teprve poté skladníci vybírají požadované množství dle požadavků zákazníků. To znamená, že kompletace objednávek probíhá ve dvou krocích, kdy nejprve je vychystána část objednávky ve skladech maloobrátkového zboží a stavební chemie, zboží je následně převezeno do kompletační zóny skladu rychloobrátkového zboží, kde je doplněna zbývající část objednávky. Kompletované palety jsou na baličce zabaleny do strečové fólie, převezeny pomocí čelních vozíků s elektrickým pohonem do expediční zóny skladu a naloženy z čela do nákladních automobilů přistavených na 2 nakládacích rampách. Veškerý pohyb manipulační techniky v regálových systémech je při kompletaci prováděn po nejkratší cestě, dále je při vychystávání dodržován systém first in, first out (FIFO), u skupiny zboží obklady vychystávání stejných barevných šarží a u skupiny zboží stavební chemie sledována doba expirace. Popsaný návrh koncepce logistiky v centrálním skladu je následně testován pomocí obecného simulačního modelu materiálových toků s cílem nalézt odpovědi na následující otázky:

- Kolik pracovníků a manipulační techniky bude třeba k realizaci materiálových toků o požadované intenzitě?
- Bude skladová kapacita umístěná v budovách dostatečná s ohledem na zavedení strategie řízení zásob pomocí hladin?
- Bude velikost logistických ploch v příjmových, kompletačních a expedičních zónách skladů dostatečná?

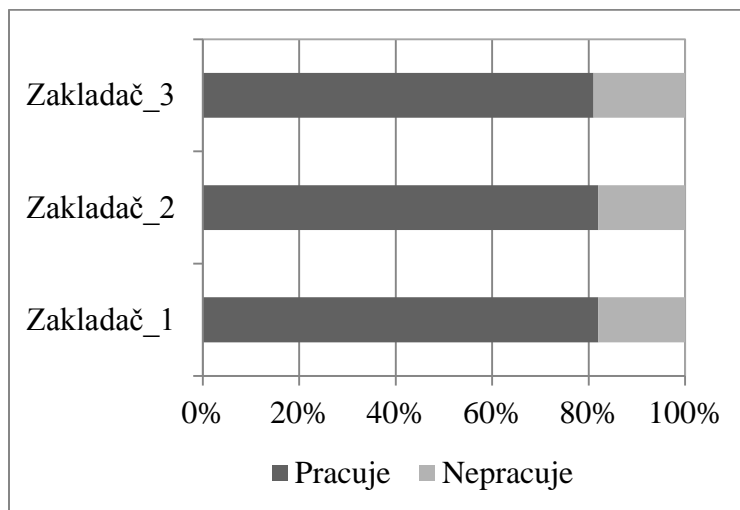
Pro nalezení odpovědi na výše uvedené otázky je pomocí obecného simulačního modelu materiálových toků modelován materiálový tok skladovým areálem rozložený do pohybů: vykládka nakoupeného zboží v příjmových zónách skladů, kontrola množství a označení zboží štítkem, zaskladnění zboží do regálů, kompletace objednávek, balení kompletované palety a nakládka zboží do nákladních automobilů při expedici. Trvání pohybů je pro navrhovaný koncept odhadnuto na základě trvání operací v současném logistickém systému a také na základě zkušeností s podobnými modelovanými systémy. Průměrná rychlost pohybu zdroje čelní vozík je uvažována 5 km/h a zdroje zakladač pro VNA technologii 7 km/h. Vstupní data do simulace pokrývají období 3 roky a jsou odvozena z historických dat o skutečné poptávce po zboží rovněž za období 3 roky. K

předpovědi poptávky je využita regresní analýza pro zachycení trendu kombinovaná s výpočtem a průměrováním indexů pro zachycení sezónnosti. Objednávky zboží jsou generovány pomocí PSMS a pro stanovení jejich výše je použita logika PQ-systému, přičemž je respektován požadavek na vyšší úrovně služeb ve formě ukazatele fill rate. Fill rate pro rychloobrátkové položky je požadována ve výši 98 %, pro položky maloobrátkové a položky ze skupiny stavební chemie ve výši 90 %. Pro zajištění pohybu pracovníků a manipulační techniky po nejkratší cestě je při simulaci pohybů představujících kompletaci objednávky využíván CWS. Pohyby tvořící materiálový tok jsou simulovány pomocí ZMS na základě požadavků na provedení pohybu generovaných generátory požadavků, které pracují s informacemi o doplňujících objednávkách získanými PSMS a také s informacemi o poptávce po zboží. Pro simulaci pohybu kompletace objednávek ve skladu stavební chemie a skladu maloobrátkového zboží je využito modifikované ZMS, kdy pro pohyb simulující kompletaci objednávek dochází k opakovanému předávání elementu pKulicka mezi elementy tvořícími ZMS mManipulaceBod1 a mPohybBod1Bod2 (viz Obr. 7.21).

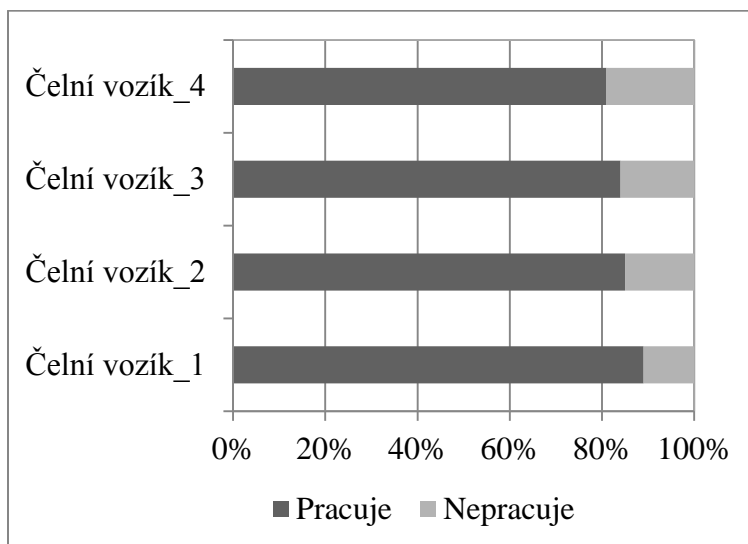


Obr. 7.21 Struktura ZMS pro simulaci pohybu kompletace objednávky ve skladu stavební chemie a skladu maloobrátkového zboží

To odpovídá situaci, kdy se skladník při kompletování objednávky ve skladu stavební chemie a maloobrátkového zboží opakovaně pohybuje po nejkratší cestě mezi lokacemi se zbožím (dynamická změna bodu 1 a bodu 2 v ZMS) až do dokončení objednávky, kdy je kompletovaná paleta převezena do kompletační zóny skladu rychloobrátkového zboží. Uvedená změna struktury ZMS je v obecné simulaci materiálových toků provedena pomocí úpravy programového kódu Output Rule elementu mPohybBod1Bod2. Výstupní pravidlo je pro pohyb kompletace objednávky podmíněno logickou strukturou IF/ELSE/ENDIF. Výstupy ze simulace pro první simulovaný rok s předpokládaným růstem tržeb ve výši 15 % mají podobu grafů zachycených na následujících obrázcích:



Obr. 7.22 Využití zakladačů pro VNA v prvním roce simulace



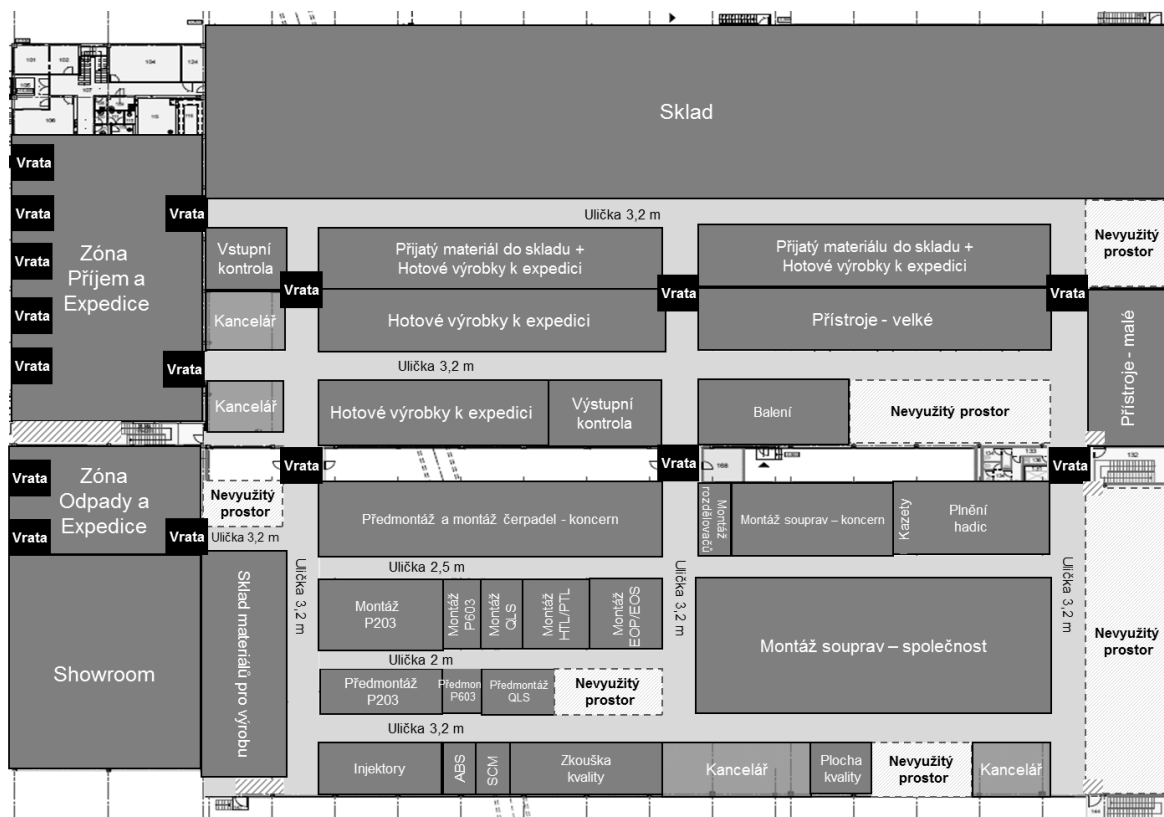
Obr. 7.23 Využití čelních vozíků v prvním roce simulace

Pro provoz navrženého logistického systému s předpokládaným růstem tržeb o 15 % za rok je dle výstupů simulace třeba 7 pracovníků na směnu, přičemž 4 pracovníci při své činnosti používají čelní vysokozdvížné vozíky a 3 pracovníci obsluhují zakladače pro VNA. Systém je provozován v režimu 5 dnů v týdnu, 3 směny. Využití skladovací kapacity se pohybuje v průměru kolem 70 % s maximem 81 %, využití logistických ploch na příjmu v průměru kolem 50 % s maximem 84 % a využití logistických ploch v expedici a při kompletaci v průměru kolem 72 % s maximem 91 %. Z toho vyplývá, že pro předpokládaný nárůst tržeb o 15 % za rok se navržené skladovací kapacity a logistické plochy jeví jako dostatečné.

7.3 Optimalizace layoutu výrobní haly a návrh systému manipulace mezi výrobou a skladem společnosti zabývající se výrobou mazacích systémů

7.3.1 Současný layout výrobní haly a systém manipulace mezi výrobou a skladem; simulace současného stavu pomocí obecného simulačního modelu materiálových toků

Společnost se zabývá sériovou výrobou mazacích systémů, které se používají při údržbě strojních zařízení. Portfolio čítá cca 4 000 hotových výrobků a polotovarů, které jsou předmětem obchodní činnosti společnosti. Tyto výrobky a polotovary se skládají z asi 9 000 dílů. Výroba a logistika probíhají v prostorách, které jsou rozděleny na zónu, ve které probíhá příjem materiálů a expedice hotových výrobků a polotovarů, dále na zónu skladu a prostory výroby (viz layout současného uspořádání haly na Obr. 7.24). Sklad disponuje VNA technologií s paletovými regály a manipulační technikou umožňující skladování do výšky 9 metrů a s celkovou kapacitou cca 5 000 paletových míst pro řízené skladování materiálů, polotovarů a hotových výrobků na 12 typech palet, které se liší svými rozměry. Řízené skladování probíhá na základě údajů o obrátkovosti položek a je zajišťováno pomocí WMS systému napojeného na informační systém společnosti. Realizaci materiálových toků ve výrobě a mezi výrobou a skladem zajišťují 2 skupiny pracovníků. Tým logistika má na starosti zásobování výrobních pracovišť vstupními materiály a polotovary a také přemísťování hotových výrobků a polotovarů z pracoviště výstupní kontroly do skladu. Manipulace s materiály a polotovary ze skladu do výroby probíhá ve 2 režimech. Materiál vychystávaný ve skladu v rámci systému řízení materiálových toků, který je založen na principu kanbanu, je manipulován tažnou soupravou se 2 vozíky. Každý vozík disponuje 3 patry vhodnými pro uložení drobného materiálu (v tomto případě především spojovacího materiálu) v plastových KLT boxech malých rozměrů. Materiál v KLT boxech je na pracovištích umísťován do spádových regálů v rámci dodržování pravidla o výměně prázdného boxu za plný.



Obr. 7.24 Současné uspořádání layoutu výrobní haly

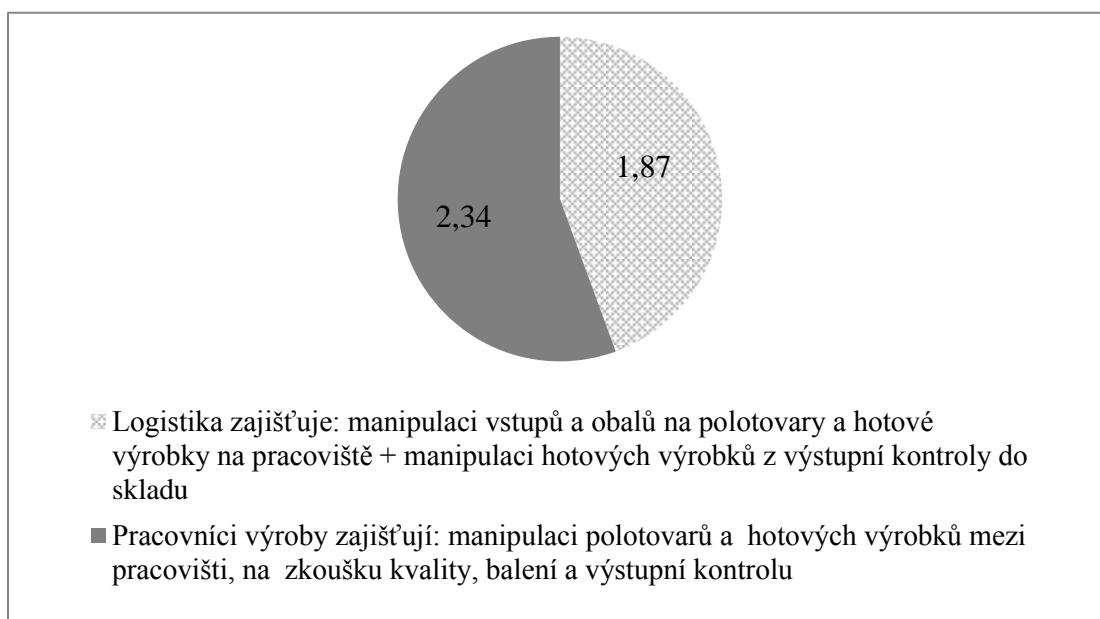
Materiál vychystávaný ve skladu přímo na výrobní zakázky je manipulován pomocí ručně vedených vozíků bez zdvihu, a to na paletách, na kterých jsou umístěny gitterboxy, sudy a plastové přepravky větších rozměrů. Hotové výrobky a polotovary jsou z pracoviště výstupní kontroly přesunovány do skladu na paletách pomocí ručně vedených vozíků bez zdvihu. Realizaci materiálových toků mezi jednotlivými pracovišti ve výrobě zajišťují přímo pracovníci těchto pracovišť, přičemž k manipulaci polotovarů umístěných na paletách využívají ručně vedených vozíků bez zdvihu. Z logistického auditu současného uspořádání layoutu výrobní haly a procesů probíhajících při realizaci materiálových toků ve výrobě a mezi výrobou a skladem vyplynuly 3 základní nedostatky týkající se rozmístění pracovišť, šířky manipulačních uliček a systému manipulace:

- Do uliček o šířce menší než 3 metry nemůže zajíždět tažná souprava. Při doplňování vstupních materiálů na bázi kanbanu zastaví pracovník logistiky obsluhující tažnou soupravu v uličce s šířkou 3,2 metru a poté chodí s boxy k jednotlivým pracovištím umístěným v úzkých uličkách a vyměňuje ve spádových regálech prázdné boxy za plné.
- Pracoviště zkouška kvality leží proti směru materiálových toků, uvažíme-li, že 95 % toků na toto pracoviště přichází z uliček o šířce 2,5 metru, a to konkrétně

z pracovišť předmontáž a montáž čerpadel – koncern, předmontáž a montáž čerpadel – společnost a montáž souprav – koncern.

- Manipulace vždy pouze 1 palety pomocí ručně vedených vozíků je neefektivní. V případě manipulací, které zajišťují pracovníci výroby, zabírá čas spotřebovaný na manipulaci čas, který by bylo možné věnovat výrobě.

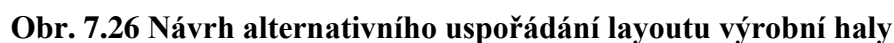
Pomocí obecného simulačního modelu materiálových toků byla dále stanovena potřeba pracovníků zajišťujících provoz současného systému manipulace (viz Obr. 7.25).



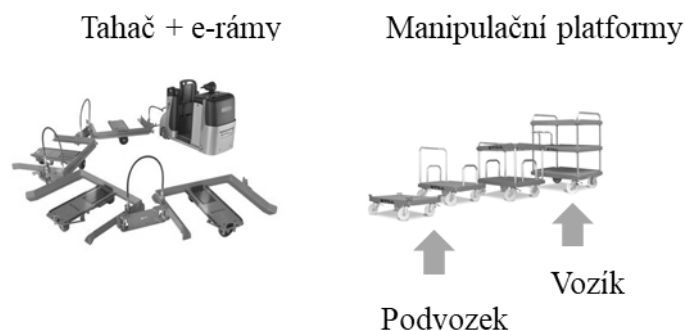
Obr. 7.25 Potřeba pracovníků k zajištění provozu současného systému manipulace

Vstupy do simulace představují historické informace o materiálových tocích mezi skladem a výrobou za období 1 roku. Simulované pohyby tvořící materiálový tok jsou manipulace vstupů a polotovarů ze skladu na pracoviště výroby, manipulace polotovarů mezi pracovišti výroby, zkouškou kvality, balením a výstupní kontrolou a manipulace hotových výrobků z pracoviště výstupní kontroly do skladu. Průměrná rychlost tažené soupravy je uvažována 5 km/h, průměrná rychlost chůze s ručně vedeným vozíkem bez zdvihu 4 km/h. Manipulace s paletami trvá průměrně 0,8 min/1 manipulace, manipulace s KLT boxy 0,4 min/ 1 manipulace, 1 box. Výroba a sklad jsou provozovány v režimu 5 dnů v týdnu, 2 směny s délkou trvání směny 8 hodin. Z výsledků simulace vyplývá, že pro provoz systému manipulace je třeba na každé směně 4,21 pracovních úvazků, přičemž 1,87 pracovních úvazků je spotřebováno na zajištění manipulací týmu logistika a 2,34 pracovních úvazků spotřebují manipulace pracovníků výroby.

Na základě výsledků logistického auditu současného uspořádání layoutu výrobní haly a procesů probíhajících při realizaci materiálových toků ve výrobě a mezi výrobou a skladem byl navržen layout alternativní (viz Obr. 7.26).

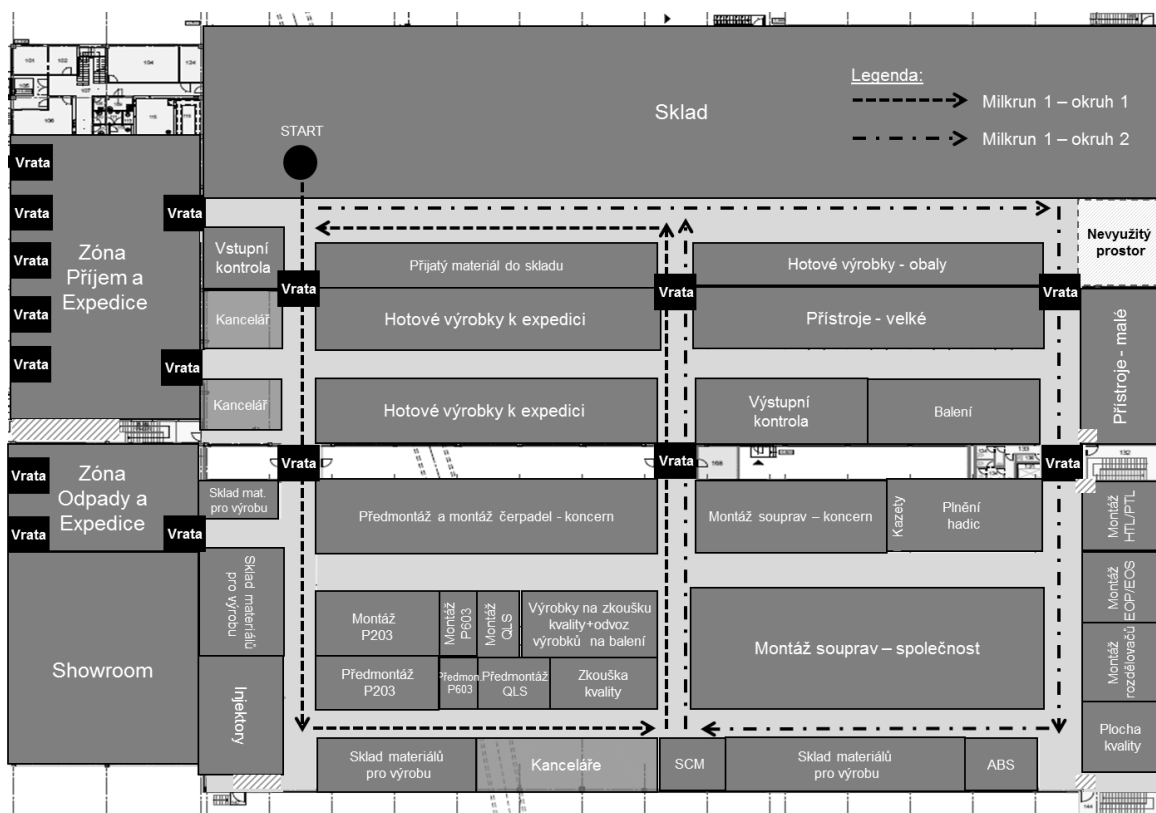


149

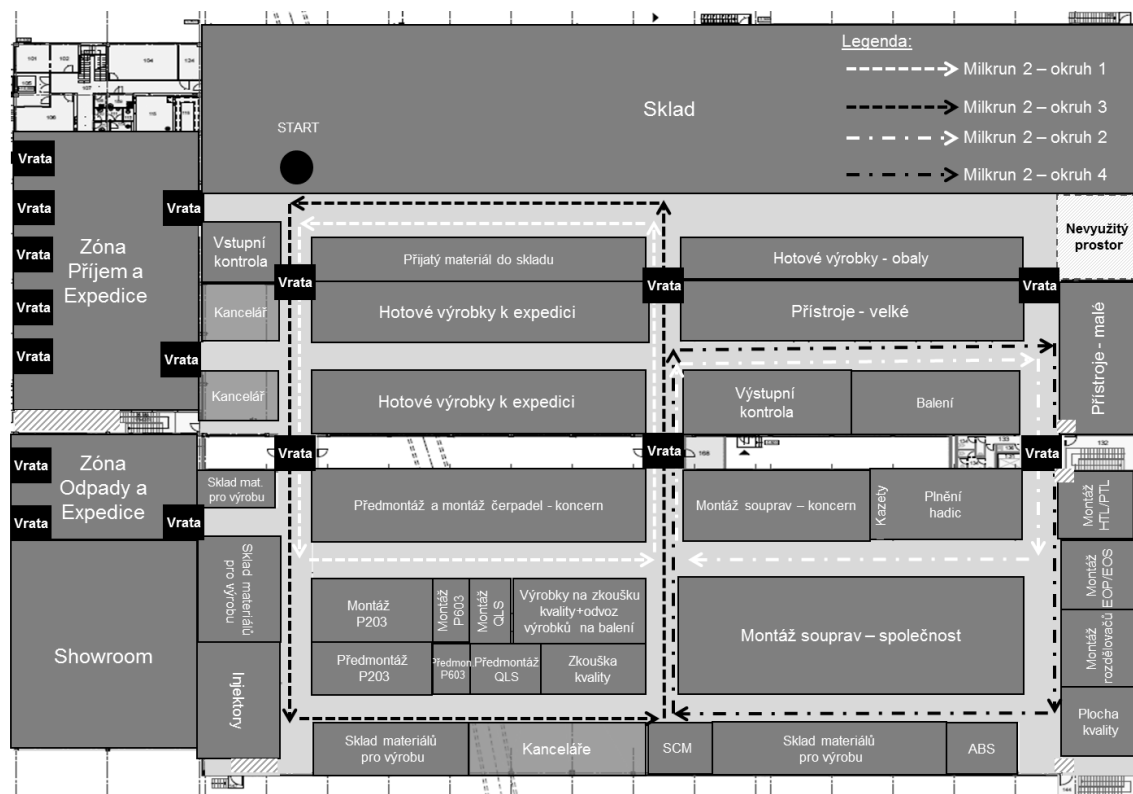


Obr. 7.27 Technologie pro provoz inteligentní tažné soupravy

Manipulační platformy jsou navrženy ve 2 základních provedeních, a to jednak jako podvozky použitelné pro manipulaci sudů, palet, gitterboxů a plastových přepravek větších rozměrů, a jednak v podobě vozíků podobných vozíkům používaným v současném systému manipulace pro manipulaci materiálů v KLT přepravekách. Inteligentní tažné soupravy jsou používány k manipulaci materiálových toků ve 2 systémech Milkrun, nazvaných Milkrun 1 a Milkrun 2. Tyto systémy se vyznačují pohybem souprav v pevně daných okruzích, které jsou pomocí šipek popsány na Obr. 7.28 a Obr. 7.29.



Obr. 7.28 Návrh okruhů systému manipulace Milkrun 1



Obr. 7.29 Návrh okruhů systému manipulace Milkrun 2

Systém Milkrun 1 je určen k zásobování výrobních pracovišť vstupními materiály a polotovary a také k přesunu hotových výrobků a polotovarů z pracoviště výstupní kontroly do skladu. Skládá se ze 2 okruhů, které začínají i končí ve skladu, přičemž během 1 jízdy je absolvován vždy pouze 1 z okruhů. Systém Milkrun 2 je určen k manipulaci polotovarů mezi pracovišti výroby, zkouškou kvality, balením a výstupní kontrolou a k manipulaci hotových výrobků a polotovarů z pracoviště výstupní kontroly do skladu. Skládá se ze 4 okruhů složených do jízdní sekvence se začátkem i koncem ve skladu. Inteligentní tažné soupravy se v systému Milkrun pohybují tak, aby byl splněn požadavek na jednosměrný provoz s výjimkou manipulační uličky sousedící s prostorem skladu, která svou šířkou umožňuje provoz obousměrný. Dále platí, že veškeré operace spojené s manipulací provádí tým logistika.

Pomocí obecného simulačního modelu materiálových toků jsou dále zjišťovány odpovědi na následující otázky:

- Kolik pracovníků, tahačů, e-rámů a manipulačních platforem je třeba k provozování systému Milkrun?
- Jak často budou tažné soupravy vyrazet na okruhy tvořící systémy Milkrun 1 a Milkrun 2 a jaká bude jízdní sekvence systému Milkrun 2?

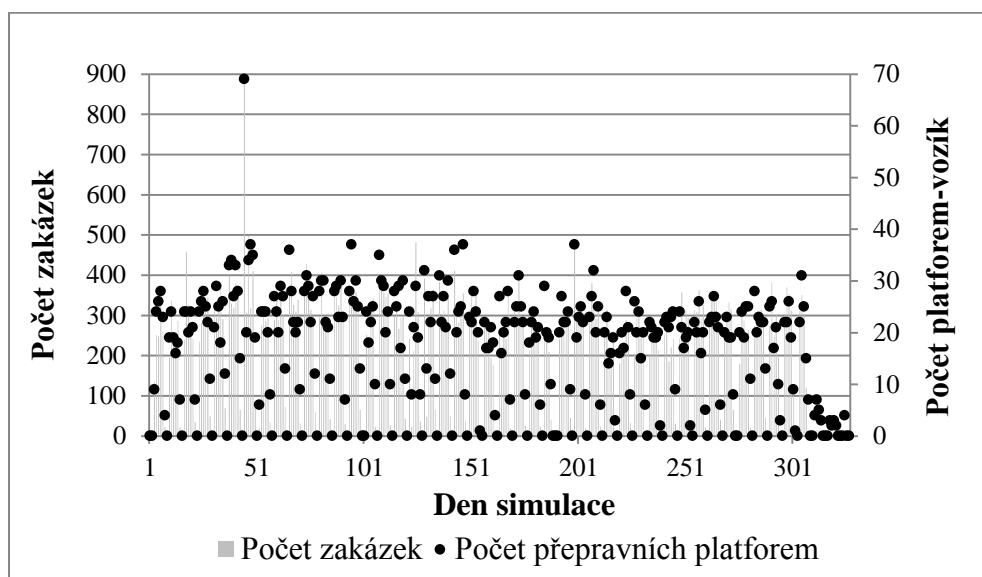
- Jakým způsobem měřit výkon a efektivitu systému manipulace Milkrun?

Vstupy do simulace představují historické informace o materiálových tocích mezi skladem a výrobou za období 1 roku, stejně jako při simulaci současného stavu. Simulované pohyby tvořící materiálový tok jsou manipulace vstupů a polotovarů ze skladu na pracoviště výroby, manipulace polotovarů mezi pracovišti výroby, zkouškou kvality, balením a výstupní kontrolou a manipulace hotových výrobků z pracoviště výstupní kontroly do skladu. Průměrná rychlost tažné soupravy je uvažována 5 km/h. Manipulace s manipulačními platformami při jejich zapojení do e-rámu trvá průměrně 0,2 min/1 zapojení, manipulace s manipulačními platformami při jejich vyjmutí z e-rámu trvá průměrně 0,45 min/1 vyjmutí a manipulace s KLT boxy 0,4 min/ 1 manipulace, 1 box. Výroba a sklad jsou provozovány v režimu 5 dnů v týdnu, 2 směny s délkou trvání směny 8 hodin. Pro simulaci uvedených pohybů jsou v obecném simulačním modelu materiálových toků využity modifikované ZMS v podobě stejné jako na Obr. 7.21. Cycle Time elementu mManipulaceBod1 se pro každý navštívený bod ležící na okruhu či jízdní sekvenci systému Milkrun skládá z času vyjmutí manipulačních platform z e-rámů, zapojení manipulačních platform do e-rámů a manipulace s KLT boxy s výjimkou bodu reprezentujícího sklad na počátku okruhu, kdy dochází pouze k zapojení manipulačních platform do e-rámů. Po navštívení všech míst, jež je simulováno předáváním elementu pKulicka mezi elementy mManipulaceBod1 a mPohybBod1aBod2, je okruh či jízdní sekvence Milkrunu ukončena opět v bodu, který reprezentuje sklad, vyjmutím manipulačních platform z e-rámů (práce elementu mManipulaceBod2).

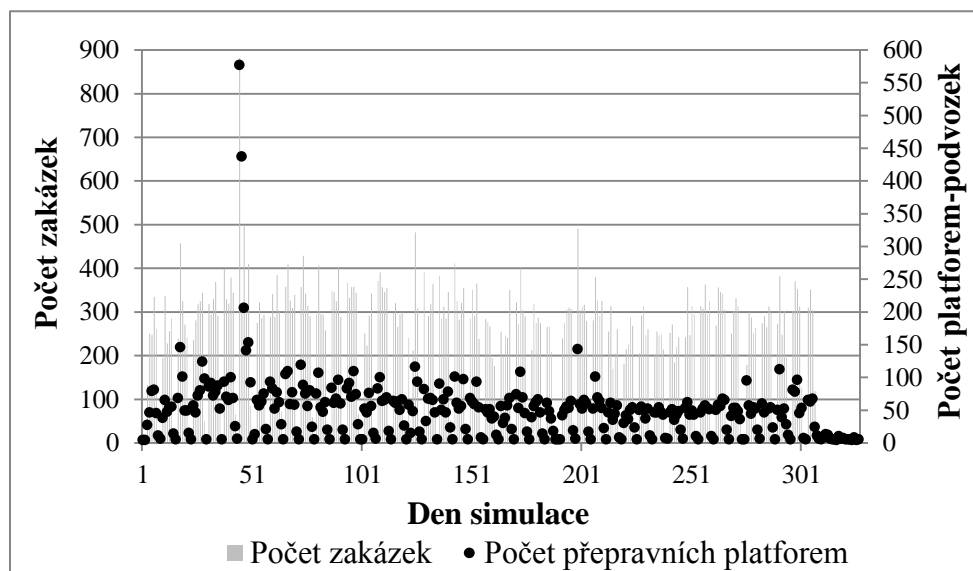
Pomocí experimentování s obecným simulačním modelem materiálových toků bylo zjištěno následující:

- Optimální počet inteligentních tažných souprav obsluhujících systém Milkrun je 2.
- První soupravu je optimální využívat výhradně pro obsluhu systému Milkrun1; druhá souprava necht' je používána jak pro obsluhu systému Milkrun1, tak pro obsluhu systému Milkrun2.
- Optimální pro zahájení jízdy soupravy obsluhující systém Milkrun1 je okamžik, kdy jsou ve skladu pro Okruh 1 či Okruh 2 z Obr. 7.28 připraveny minimálně 3 manipulační platformy se vstupními materiály a polotovary určenými pro pracoviště výroby nebo kdy na pracovišti výstupní kontrola čekají minimálně 3 manipulační platformy s hotovými výrobky a polotovary na odvoz do skladu.
- Optimální jízdní sekvence systému Milkrun2 je dána posloupností okruhů Okruh 1 – Okruh 2 – Okruh 2 – Okruh 1 – Okruh 4 – Okruh 3 z Obr. 7.29.
- Souprava obsluhující systém Milkrun2 necht' vyráží na jízdní sekvenci tvořenou okruhy jednou za 35 minut.

Potřebu manipulačních platforem pro zajištění manipulace materiálových toků zachycují grafy na Obr. 7.30 a Obr. 7.31.



Obr. 7.30 Potřeba manipulačních platforem – vozík

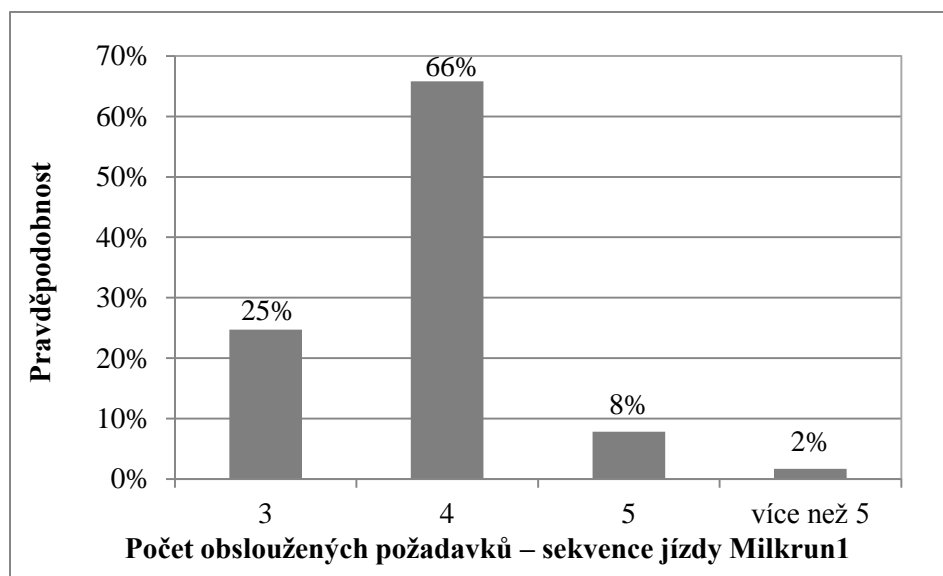


Obr. 7.31 Potřeba manipulačních platforem – podvozek

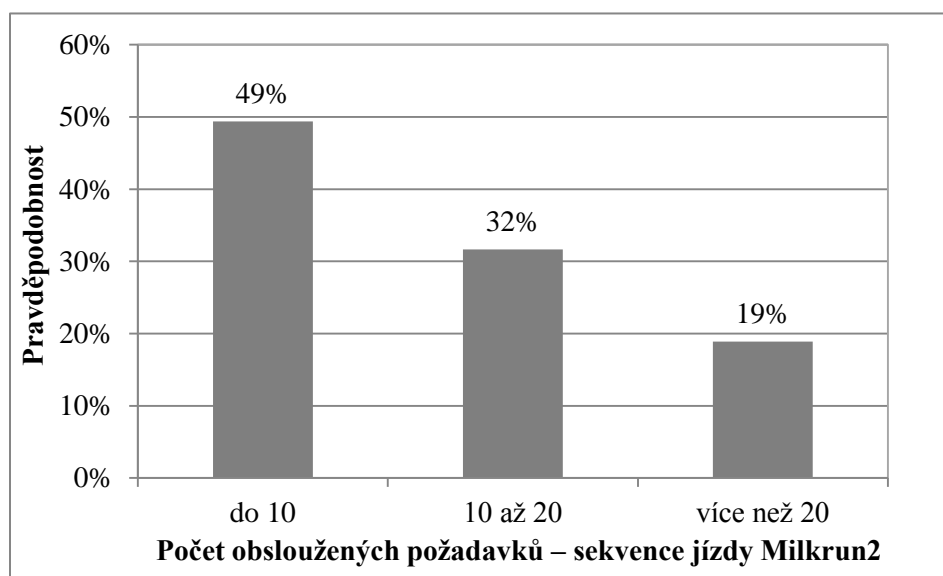
V grafech jsou pro každý simulovaný den uvedeny počty výrobních zakázek a maximální počet manipulačních platforem typu vozík a podvozek, jež jsou využity při manipulaci spojené s těmito zakázkami. Pro manipulační platformu typu vozík platí, že v 90 % simulovaných dnů bylo třeba maximálně 30, v 95 % simulovaných dnů 35 a v 98 %

simulovaných dnů maximálně 40 kusů platformy. Pro manipulační platformu typu podvozek platí, že v 90 % simulovaných dnů bylo třeba maximálně 100, v 95 % simulovaných dnů 120 a v 98 % simulovaných dnů maximálně 140 kusů platformy.

Pro posouzení efektivity systému manipulace Milkrun jsou v grafech na Obr. 7.32 a Obr. 7.33 sledovány počty požadavků, jež jsou obslouženy během 1 jízdy inteligentní tažné soupravy.

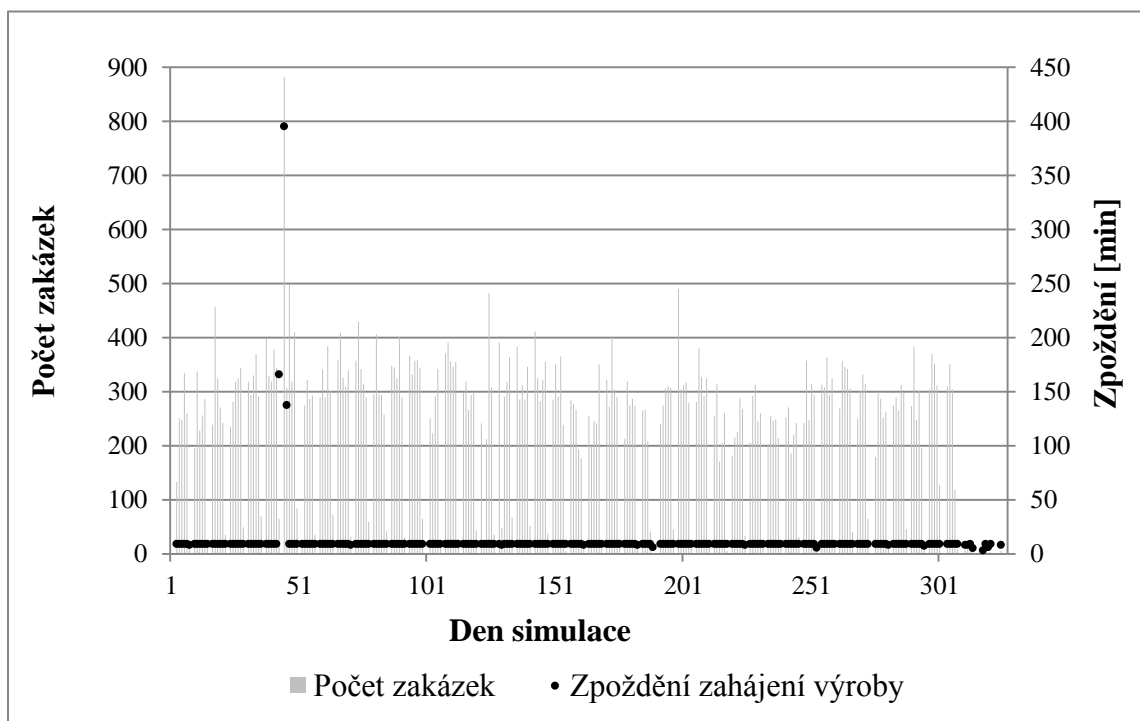


Obr. 7.32 Počet obslužených požadavků během jedné jízdy – Milkrun 1



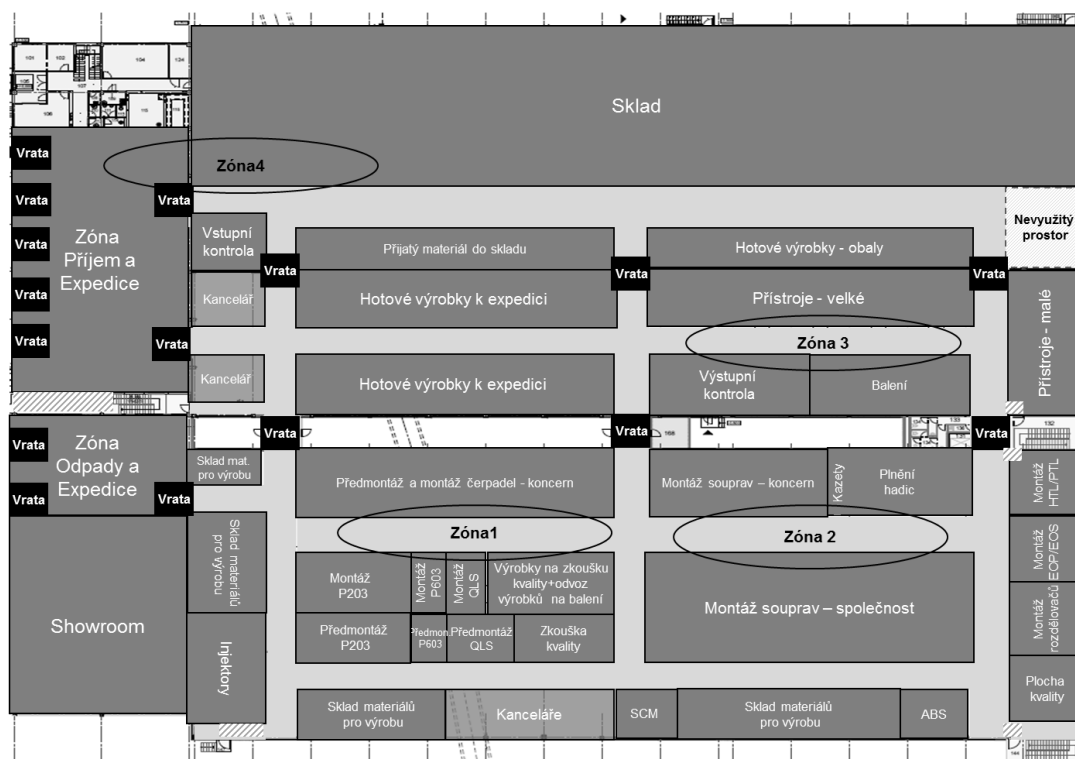
Obr. 7.33 Počet obslužených požadavků během jedné jízdy – Milkrun 2

Dále je na Obr. 7.34 uvedeno maximální zpoždění zahájení výroby zakázky pro každý simulovaný den. Toto zpoždění je způsobeno tím, že tažné soupravy obsluhující systém Milkrun 1 a Milkrun 2 nestačí dodávat potřebné množství vstupních materiálů a polotovarů na pracoviště výroby tak, aby výroba probíhala dle plánu. Platí, že pro 99,7 % zakázek se maximální doba zpoždění zahájení výroby pohybuje do 10 minut.

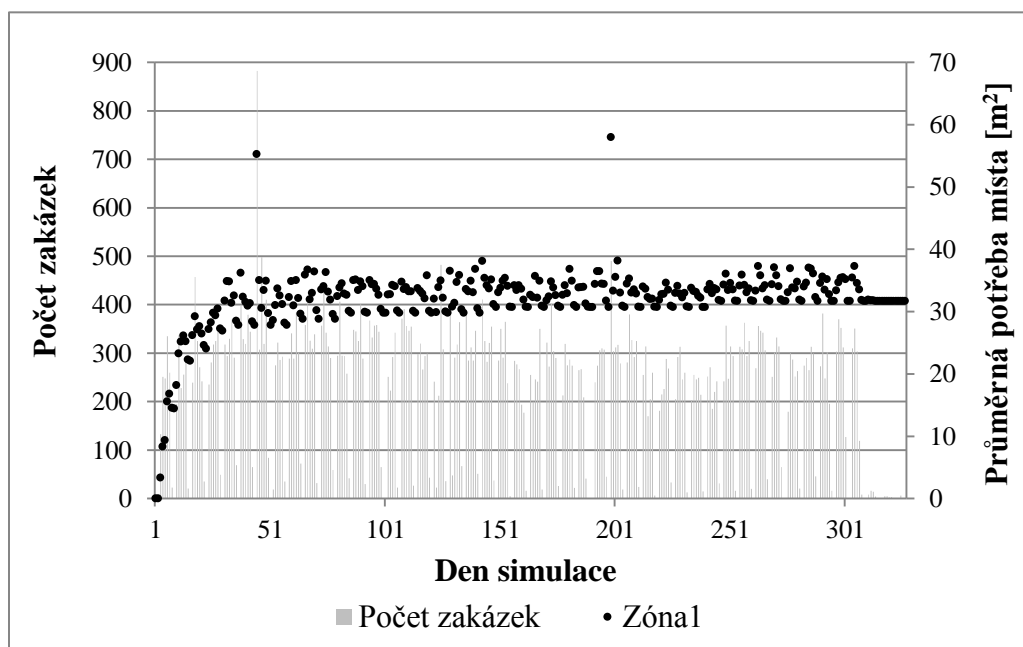


Obr. 7.34 Zpoždění zahájení výroby zakázky

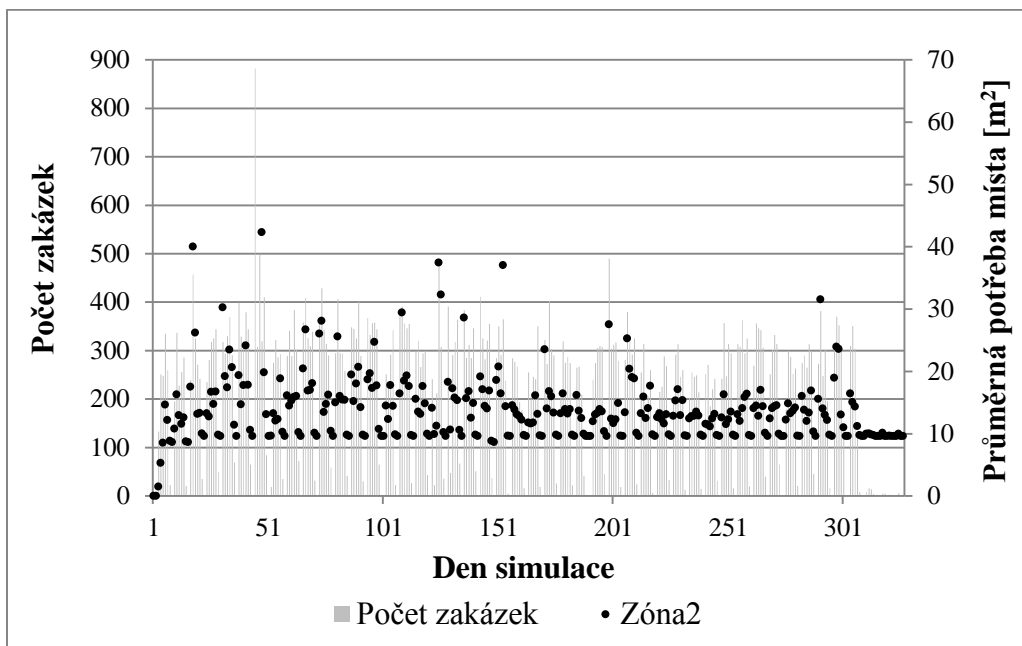
Pro zjištění potřeby místa v částech haly, kde dochází k největší koncentraci materiálu (viz zóny 1–4 v layoutu na Obr. 7.35), slouží výstupy ze simulace v podobě grafů na Obr. 7.36, Obr. 7.37, Obr. 7.38 a Obr. 7.39. V grafech jsou pro každý simulovaný den uvedeny počty výrobních zakázek a průměrná potřeba místa nutného pro umístění manipulačních platform. Pro zónu 1 platí, že v případě 99 % simulovaných dnů bylo třeba maximálně 38 m² logistické plochy. Pro zónu 2 platí, že v případě 99 % simulovaných dnů bylo třeba maximálně 42 m² logistické plochy. Pro zónu 3 platí, že v případě 99 % simulovaných dnů bylo třeba maximálně 35 m² logistické plochy. Pro zónu 4 platí, že ve více než 99 % simulovaných dnů bylo třeba maximálně 29 m² logistické plochy.



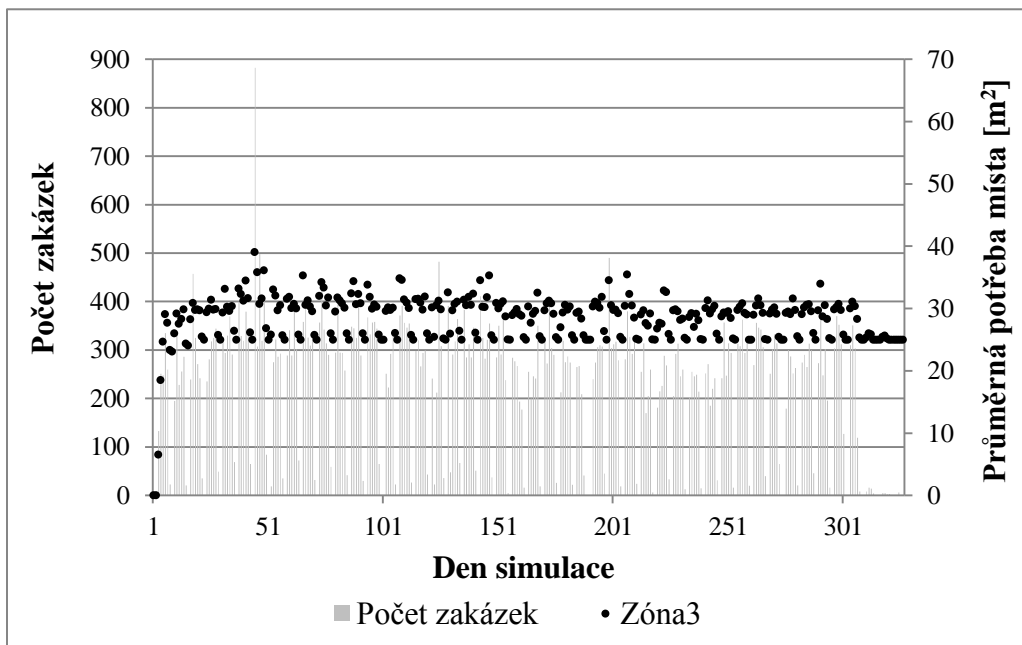
Obr. 7.35 Vymezení zón pro stanovení potřebné velikosti logistických ploch



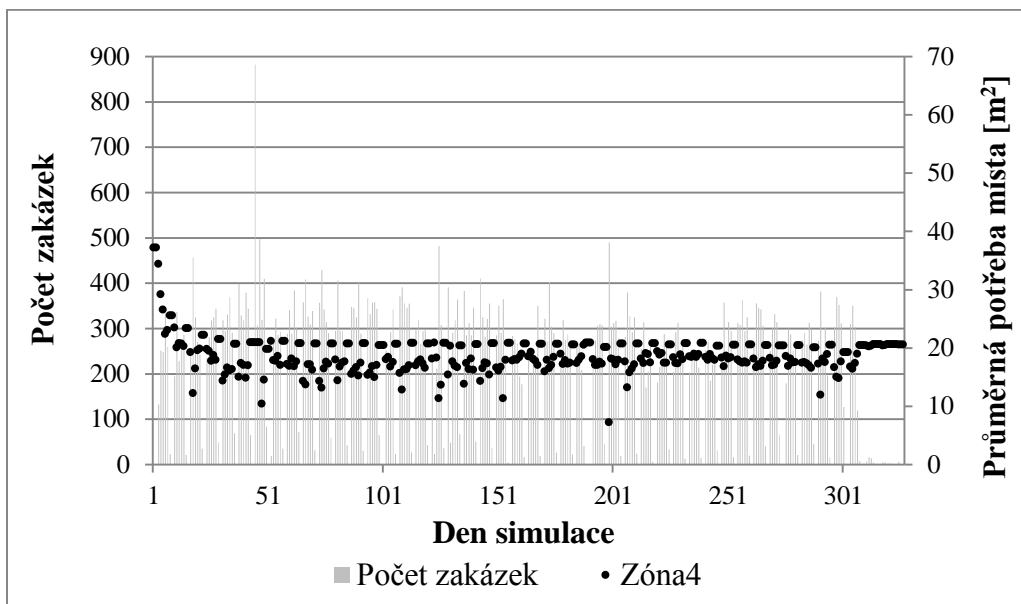
Obr. 7.36 Potřeba místa – Zóna 1



Obr. 7.37 Potřeba místa – Zóna 2

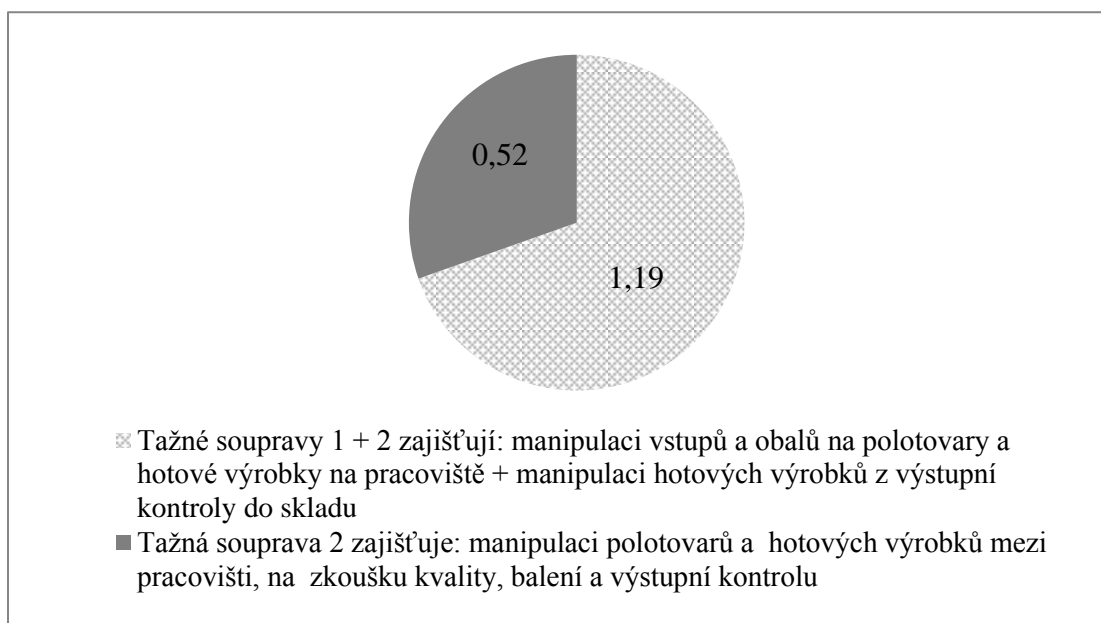


Obr. 7.38 Potřeba místa – Zóna 3



Obr. 7.39 Potřeba místa – Zóna 4

Stejně jako v případě simulace současného stavu byla pro alternativní návrh layoutu stanovena potřeba pracovníků, kteří obsluhují systém Milkrun (viz Obr. 7.40).



Obr. 7.40 Potřeba pracovníků k zajištění provozu systému manipulace Milkrun

Z výsledků simulace vyplývá, že pro provoz systému manipulace Milkrun je třeba na každé směně 1,71 pracovních úvazků. Při porovnání se současným stavem došlo ke snížení

potřeby úvazků nutných k zajištění manipulace se vstupními materiály a polotovary ze skladu na pracoviště a k manipulaci s hotovými výrobky z výstupní kontroly do skladu z 1,87 na 1,19. V případě přesunu polotovarů a hotových výrobků mezi pracovišti došlo k poklesu potřebných pracovních úvazků z 2,34 na 0,52. Celková úspora pracovních úvazků v porovnání se současným stavem je 2,5 pracovních úvazků na směnu.

7.4 Reengineering procesů spojených s vychystáním objednávky zákazníka v centrálním skladu společnosti distribuující spotřební zboží

7.4.1 Současný stav logistiky v centrálním skladu společnosti

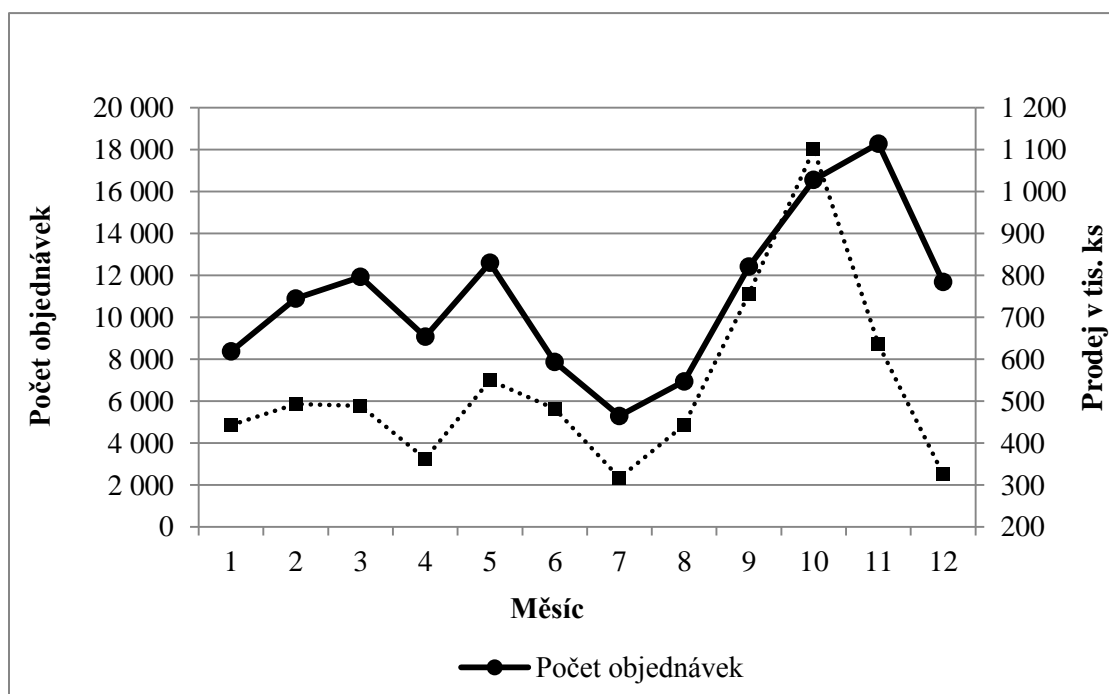
Společnost se zabývá distribucí spotřebního zboží na území České republiky. Jedná se o portfolio, které tvoří cca 12 000 aktivních položek. Kusová hmotnost distribuovaných položek je maximálně 2 kilogramy a každý kus položky je označen unikátním čárovým kódem typu EAN (tj. European Article Number). Distribuční systém je tvořen centrálním skladem umístěným v Česku v okolí Prahy. Centrální sklad společnosti je zásobován 12 výrobci, jejichž výrobní závody leží v Česku, na Slovensku a v Polsku. Tito výrobci dodávají zboží do centrálního skladu v podobě homogenních palet s maximálním množstvím položky na paletě ve výši 500 kusů. Zákazníky společnosti, kterých je v Česku asi 58 000, lze rozdělit do 7 distribučních kanálů. Charakteristiky distribučních kanálů na základě prodeje v naturálních jednotkách a počtů realizovaných objednávek ukazuje následující tabulka:

Tabulka 7.2 Charakteristiky distribučních kanálů na základě prodeje v naturálních jednotkách a počtů realizovaných objednávek

Distribuční kanál	Podíl na celkovém počtu objednávek	Podíl na celkovém objednaném množství
1	32 %	2 %
2	méně než 1 %	méně než 1 %
3	14 %	9 %
4	1 %	12 %
5	méně než 1 %	8 %
6	52 %	68 %
7	méně než 1 %	méně než 1 %

Z hlediska logistiky je velmi náročný především distribuční kanál číslo 1, který se významně podílí na celkovém počtu objednávek, ale objednávaná množství jsou relativně nízká. Naopak distribuční kanály 4 a 5 jsou charakteristické relativně nízkým počtem objednávek, přičemž objednávaná množství jsou relativně velká.

Další charakteristika distribučního systému zohledňuje rozložení poptávky v čase a je uvedena na následujícím obrázku:



Obr. 7.41 Rozložení poptávky v čase

Rozložení poptávky v čase ukazuje výrazný sezónní výkyv v měsících 10 a 11. V měsíci 10 je sezónní výkyv způsoben především poptávkou v distribučním kanálu 6, a proto se projevuje jak zvýšeným počtem objednávek, tak zvýšenými prodeji v naturálních jednotkách. V měsíci 11 je sezónní výkyv reprezentován pouze zvýšeným počtem objednávek, který jde na vrub distribučnímu kanálu číslo 1. Naopak v měsících 6–8 dochází k útlumu poptávky, jde tedy o období vhodné pro provedení případných změn ve skladové logistice.

Podrobnější pohled na zvyklosti zákazníků z hlediska rozložení objednávek během týdne (tj. ve kterých dnech a v jakých časech zákazníci nejčastěji objednávají) je uveden v následující tabulce:

Tabulka 7.3 Zvyklosti zákazníků z pohledu rozložení objednávek během týdne

Hodina/Den týdne	1	2	3	4	5	6	7	Celkem
0	0 %	0 %	0 %	0 %	0 %	0 %	0 %	1 %
1	0 %	0 %	0 %	0 %	0 %	0 %	0 %	1 %
2	0 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %
3	0 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %

4	0 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %
5	0 %	0 %	1 %	0 %	0 %	0 %	0 %	2 %
6	0 %	0 %	1 %	0 %	0 %	0 %	0 %	2 %
7	0 %	0 %	1 %	0 %	0 %	0 %	0 %	2 %
8	1 %	1 %	1 %	1 %	1 %	0 %	0 %	5 %
9	1 %	1 %	2 %	1 %	1 %	0 %	0 %	7 %
10	1 %	2 %	2 %	1 %	1 %	0 %	0 %	8 %
11	2 %	2 %	2 %	1 %	1 %	0 %	0 %	8 %
12	2 %	1 %	2 %	1 %	1 %	0 %	0 %	7 %
13	2 %	1 %	2 %	1 %	1 %	0 %	0 %	8 %
14	2 %	2 %	2 %	1 %	1 %	0 %	0 %	8 %
15	1 %	1 %	1 %	1 %	1 %	0 %	0 %	7 %
16	1 %	1 %	1 %	1 %	1 %	0 %	0 %	6 %
17	1 %	1 %	2 %	1 %	1 %	0 %	0 %	6 %
18	1 %	1 %	2 %	1 %	0 %	0 %	0 %	5 %
19	1 %	1 %	1 %	0 %	0 %	0 %	1 %	4 %
20	1 %	1 %	1 %	1 %	0 %	0 %	1 %	4 %
21	1 %	1 %	1 %	0 %	0 %	0 %	1 %	3 %
22	0 %	1 %	0 %	0 %	0 %	0 %	0 %	3 %
23	0 %	0 %	0 %	1 %	0 %	0 %	0 %	2 %
Celkem	18 %	18 %	23 %	16 %	14 %	4 %	6 %	

Z této tabulky je zřejmé, že zákazníci nejčastěji objednávají ve středu v časech od 8 do 18 hodin.

Charakteristiky distribučních kanálů 1, 3, 4, 5 a 6 z hlediska počtu položek v jedné objednávce a objednaného množství položky v kusech zobrazují následující tabulky:

Tabulka 7.4 Počet položek v jedné objednávce

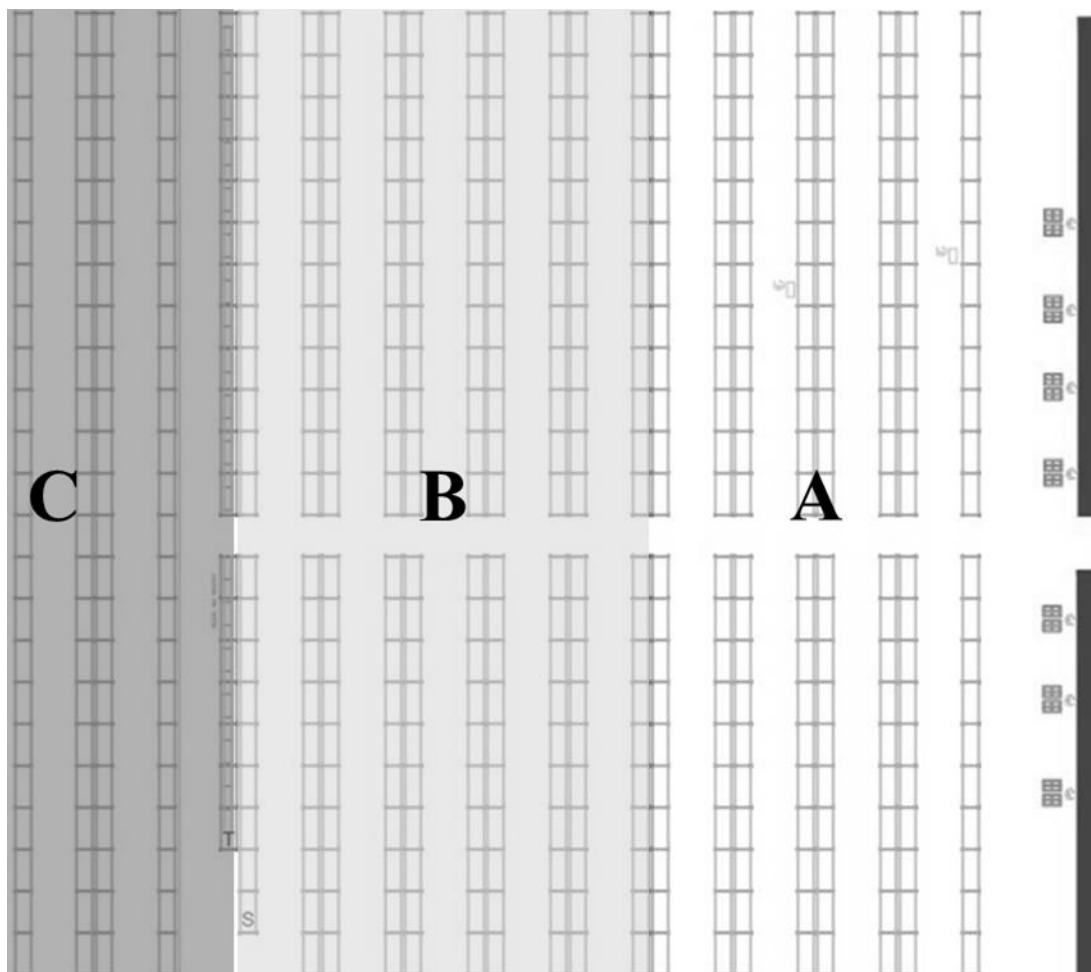
Kvantil/Distribuční kanál	1	3	4	5	6
10 %	1	2	1	1	1
20 %	1	4	2	1	3
30 %	1	7	3	1	5
40 %	1	11	4	2	7
50 %	2	16	6	3	9
60 %	2	21	8	5	12
70 %	3	27	12	8	15
80 %	4	36	19	15	22
90 %	6	48	30	15	34
95 %	8	59	47	23	49

Tabulka 7.5 Objednaná množství položky v kusech

Kvantil/Distribuční kanál	1	3	4	5	6
10 %	1	3	7	10	5
20 %	1	6	53	26	10
30 %	1	9	214	30	13
40 %	1	14	380	40	16
50 %	2	20	497	100	21
60 %	2	27	535	230	27
70 %	3	36	595	536	38
80 %	4	49	600	1 319	56
90 %	6	71	975	2 753	100
95 %	9	95	2 469	7 530	179

Pro zákazníky z distribučního kanálu 1 je typické časté objednávání pouze 1 položky (až 40 % objednávek) v množství 1 kus. To představuje 16 % objednávek tohoto distribučního kanálu a cca 5 % z celkového počtu objednávek všech distribučních kanálů. Naopak pro distribuční kanály 4 a 5 platí, že na objednávkách se objevuje relativně malý počet položek (medián distribučního kanálu 4 je 6 a medián distribučního kanálu 5 je 3), přičemž objednávaná množství jsou řádově vyšší než u ostatních distribučních kanálů (mediány distribučních kanálů 1, 3 a 6 se pohybují maximálně v desítkách kusů a mediány distribučních kanálů 4 a 5 ve stovkách kusů). Zákazníci jsou bez ohledu na distribuční kanál rozděleni do skupin dle celkových prodejů v peněžních jednotkách. Nejvýznamnější zákazníci patřící do skupiny A se podílejí na tržbách společnosti z 80 % a je jim garantován termín dodání do 1 dne od objednání, pokud objednají ve všední den nejpozději do 16 hodin. Zákazníci patřící do skupiny B se podílejí na tržbách společnosti z 15 % a je jim garantován termín dodání do 2 dnů, pokud objednají ve všední den nejpozději do 16 hodin. Konečně zákazníci patřící do skupiny C se podílejí na tržbách společnosti z 5 % a je jim garantován termín dodání nejpozději do týdne od objednání.

Objednávky zákazníků jsou vychystávány a expedovány ze skladu, jehož provozní doba je pondělí až pátek ve dvou směnách. Délka směny je 8 hodin a směny se částečně překrývají. Sklad je vybaven regálovým systémem a má následující layout:



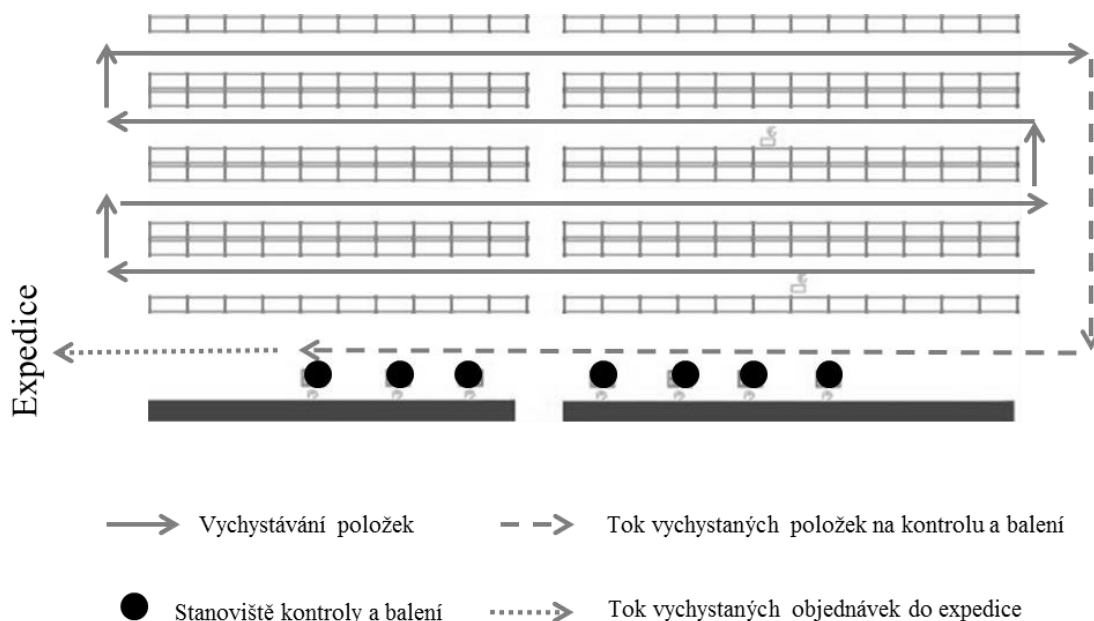
Obr. 7.42 Layout skladu a rozdělení vychystávacích pozic dle obrátkovosti

Regálový systém je kombinací míst pro skladování palet a pozic, které mají police. Paletová místa na zemi jsou určena pro vychystávání objednaných položek, které jsou obrátkové. Z polic (umístěných taktéž na zemi) se vychystávají položky s malou obrátkovostí. Rozdělení vychystávacích míst dle obrátkovosti položek je provedeno pomocí Paretova pravidla [viz Pareto and Page (1971)], kdy za položky s největší obrátkovostí (tj. položky skupiny A) jsou považovány položky, které zákazníci objednávají nejčastěji. Veškerá paletová místa od první úrovně výš slouží ke skladování zboží a jsou z nich doplňována vychystávací místa. Každé vychystávací i skladovací místo je označeno čárovým kódem, který obsahuje informaci o jeho umístění (tj. skladovou lokaci). Tato informace se skládá z označení regálové řady, regálového sloupce a úrovně. Skladová lokace je párována s čárovým kódem zboží při jeho zaskladnění během příjmu. Fyzické vychystání objednávky se skládá z následujících procesů:

- vychystávání položky,

- kontrola,
- balení,
- expedice.

Umístění procesů a tok vychystaných objednávek v layoutu skladu zachycuje následující obrázek:



Obr. 7.43 Umístění procesů a tok vychystaných objednávek v layoutu skladu

Vychystávání položek provádí operátor logistiky na základě vytištěných čárových kódů přiřazených objednávkám zákazníků při jejich převodu z informačního systému společnosti do warehouse management systému (WMS), který řídí sklad. Jedné objednávce je přiřazen vždy jeden čárový kód, ale počet objednávek, které bude operátor logistiky vychystávat najednou (tzn. počet vytištěných čárových kódů), je předmětem volby. Znamená to, že operátor logistiky má po přihlášení do WMS pomocí aplikace ve čtečce čárových kódů možnost zvolit si vychystávání více než jedné objednávky najednou, což je výhodné například v situaci, kdy dojde ke kumulaci objednávek s relativně malým počtem položek a relativně malým počtem objednaných kusů. Typicky jde o nahromadění objednávek od zákazníků v distribučním kanálu 1 v období špičky. Fyzické vychystávání položek v objednávkách se skládá z chůze mezi skladovými lokacemi, vybíráním zboží na skladových lokacích, skenování čárových kódů vybíraných položek, skladových lokací a objednávek, potvrzení odebraného množství a manipulace vychystaného zboží na stanoviště balení a kontroly po dokončení vychystávání všech zvolených objednávek.

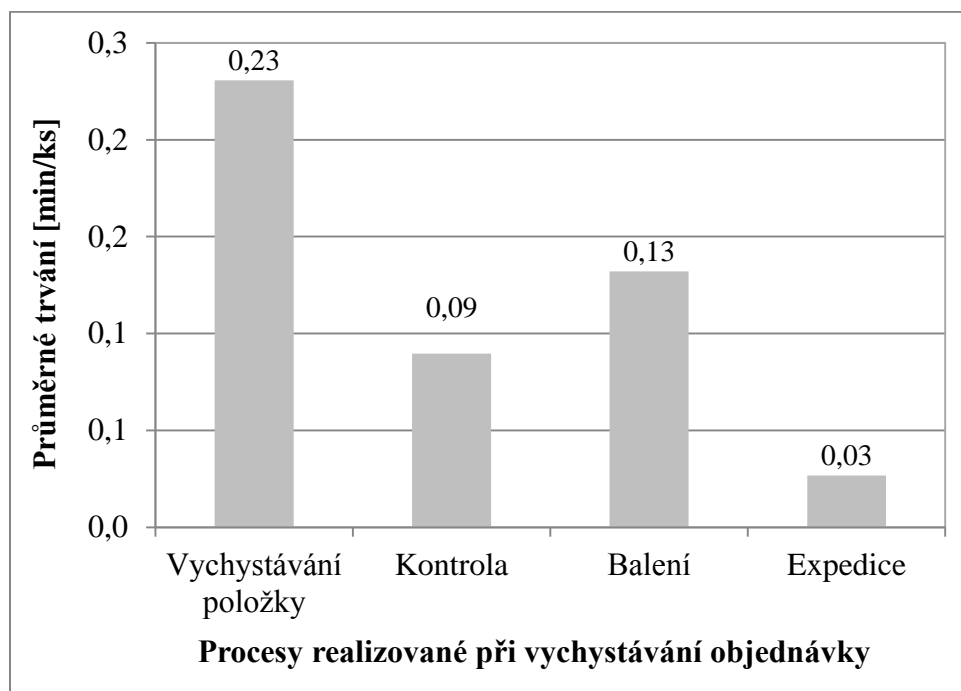
Chůze mezi lokacemi je řízena WMS na principu „bee line“ (tj. pohyb po nejkratší cestě mezi dvěma skladovými lokacemi). Zboží vybírané na skladových lokacích je ukládáno do vozíku používaného v supermarketech nebo, v případě velkých objednávek, na paletu manipulovanou ručně vedeným paletovým vozíkem.

Kontrola navazuje na proces vychystávání položek. Operátor logistiky skenuje čárový kód objednávky a poté postupně čárový kód každého kusu každé položky. V případě, že počty kusů nějaké položky jsou větší než 1, může operátor provést vizuální kontrolu a zadat počet zkontrolovaných kusů do informačního systému společnosti ručně, bez nutnosti skenovat každý kus samostatně. V případě, že kontrola prokáže správnost vychystání objednávky, potvrdí operátor logistiky tuto skutečnost v informačním systému, vytiskne příslušný počet faktur, sloužících zároveň jako dodací list, a 1 exemplář umístí do archivu společnosti.

Balení navazuje na proces kontroly. Zkontrolované položky balí operátor logistiky dle počtu objednaných kusů, buď do dvouvrstvé vlnité papírové lepenky a balicího papíru (v případě malého množství objednaných položek a množství), či do papírových krabic různých rozměrů, které vyplňuje výplňovým materiálem typu Flo-pak. Volba obalu závisí na zkušenostech operátora. Po zabalení položek přiloží operátor fakturu (je-li požadována) a obal uzavře lepicí páskou. Dále umístí na každý obal unikátní štítek dopravce a v některých případech také štítek s dodací adresou konečného zákazníka. Proces balení končí skenováním štítků dopravců, které dopravci poskytuje informaci, kolik balíků bude dopravováno, a umístěním balíků na palety či do klecových kontejnerů.

Vychystávání objednávek je zakončeno balením palet s balíky do plastové fólie a jejich nakládkou do nákladních automobilů dopravců v prostoru expedice. Balení palet zajišťuje automatická balička, manipulaci na balení a nakládku operátor logistiky.

Na základě logistického auditu byly pomocí techniky snímkování operace [viz např. Vávrová (2007)] měřeny časy trvání procesů realizovaných při vychystávání objednávky. Snímky operace byly získány v průběhu 10 směn provozu skladu v období podzimní špičky, vždy pro 8 různých operátorů logistiky. Časové náměry byly dále porovnány s informacemi o aktivitě operátorů logistiky získanými exportem z WMS, který pokrýval období 4 směn mimo špičku a 4 směn ve špičce. Na základě těchto údajů byl sestaven výkonový profil procesu vychystávání objednávky, který má následující podobu:



Obr. 7.44 Výkonový profil vychystávání objednávky

Úzkým místem při vychystávání objednávky je proces vychystávání položky. V následující tabulce je zachycen procentuální rozpad procesů tvořících výkonový profil na jednotlivé činnosti:

Tabulka 7.6 Rozklad procesů tvořících výkonový profil na činnosti

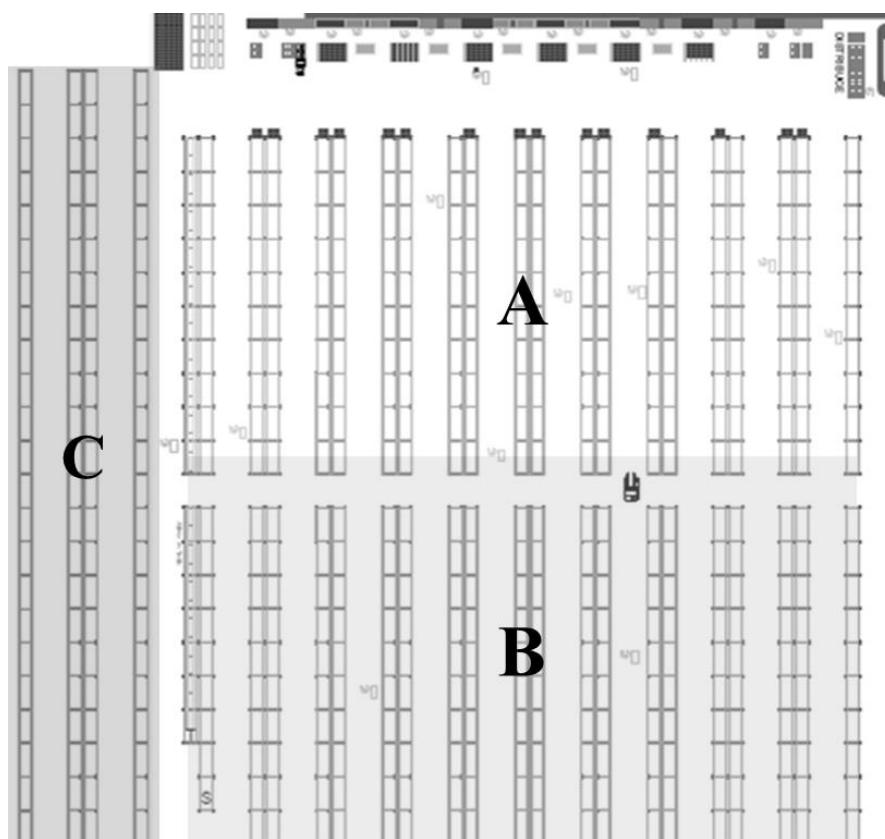
Proces	Činnost	% z trvání procesu
Vychystávání položek	Chůze	53 %
	Vybírání zboží na skladových lokacích a skenování čárových kódů	47 %
Kontrola	Skenování čárových kódů položek	31 %
	Práce v informačním systému	69 %
Balení	Balení	100 %
Expedice	Fixace palet s balíky plastovou fólií	64 %
	Nakládka palet do auta dopravce	36 %

Vzhledem k tomu, že vychystávání položek je nejpomalejším a nejdražším procesem při vychystávání objednávky, bylo primární snahou maximálně redukovat jeho trvání. Pokud by se reengineering uvedeného procesu opíral o redukci trvání činností spojených s vybíráním zboží na skladovacích lokacích a skenováním čárových kódů, bylo by nutné vynaložit značnou investici do moderních technologií (např. skenování pomocí prstu či

technologie hlasového vychystávání). Proto byla pozornost zaměřena nejprve na minimalizaci času, který skladový operátor stráví chůzí.

7.4.2 Změna systému vychystávání položek, reorganizace skladu

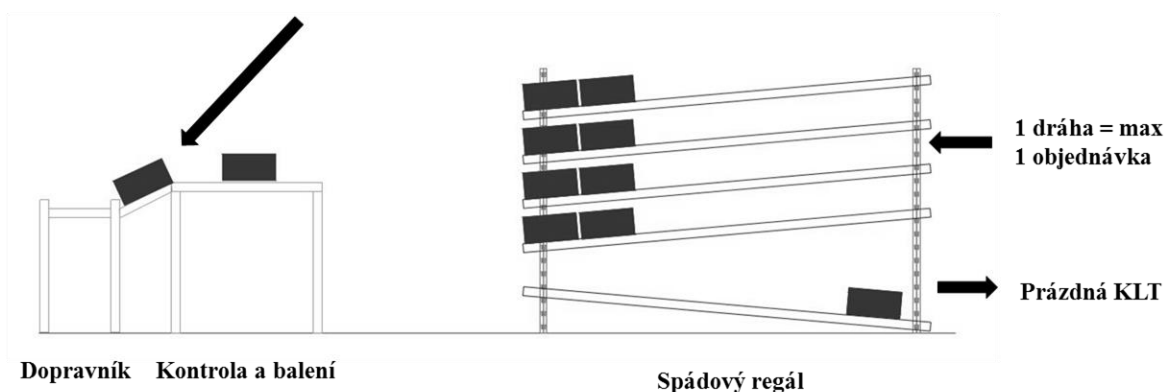
Změna systému vychystávání položek a reorganizace skladu byla navržena primárně s cílem minimalizace času, který skladový operátor stráví chůzí. Zatímco původní systém vychystávání objednávky pracoval s předpokladem, že skladový operátor logistiky vychystává vždy všechny položky na objednávce tak, že prochází teoreticky všechny uličky v regálovém systému, nový systém je založen na předpokladu, že operátor logistiky vybírá zboží na skladových lokacích vždy pouze v maximálně 2 vzájemně sousedících uličkách. To znamená, že operátor logistiky velmi pravděpodobně vychystá položky pouze části objednávky, neboť mix položek na objednávce je náhodný a položky mohou být teoreticky alokovány v jakékoli části skladu. Uvedená změna systému s sebou přináší problém, jakým způsobem spojit částečně vychystané objednávky před provedením procesů kontrola, balení a expedice. Z tohoto důvodu byla provedena reorganizace skladu, kterou lze popsat pomocí layoutu zachyceného na následujícím obrázku:



Obr. 7.45 Layout skladu po provedení reorganizace

V reorganizovaném skladu jsou přesunuta stanoviště procesů kontrola a balení do čela řad regálového systému a prostor, ve kterém se tato pracoviště nacházela původně, je zaplněn novými regály. Zároveň jsou pracoviště kontroly a balení spojena pásovým dopravníkem určeným pro automatické přemístění balíků s prostorem ukládání zásilek na palety či do klecových kontejnerů dopravců. Z důvodu možné kumulace balíků v tomto prostoru je dopravník zakončen nekonečnou smyčkou. Velmi důležitým prvkem jsou spádové regály umístěné před stanovišti kontroly a balení, ve kterých dochází ke spojování částí vychystaných objednávek. Detail spádového regálu a pracoviště, na kterém probíhají procesy balení a kontroly, zachycuje následující obrázek:

Zkontrolovaná a zabalená objednávka na dopravník



Obr. 7.46 Detail pracoviště balení a kontroly, umístění spádového regálu

Položky vychystané operátory logistiky do plastových přepravek typu KLT označených čárovým kódem jsou umisťovány do drah spádového regálu rovněž označených čárovým kódem tak, aby v jedné dráze byla umístěna vždy pouze jedna objednávka. V případě, že jsou do spádového regálu umístěny všechny vychystané položky tvořící objednávku, rozsvítí se na konci všech drah, ve kterých je objednávka umístěna světlo, které je znamením pro operátora logistiky, aby začal s realizací procesů kontroly a balení. Vždy v okamžiku vyprázdnění KLT umístí operátor provádějící kontrolu a balení přepravku do dráhy s protisměrným skluzem, čímž umožní její využití při dalším vychystávání položek.

7.4.3 Stanovení efektivity navrženého systému vychystávání položek pomocí obecného simulačního modelu materiálových toků

Z předchozího popisu vyplývá, že při vychystávání položek v novém systému absolvuje skladový operátor s manipulačním vozíkem chůzi trasy v podobě okruhu, který zahrnuje cestu operátora na začátek uličky regálového systému, ve které má začít vychystávání položek, průchod maximálně 2 sousedními uličkami regálového systému a cestu okolo spádových regálů, do kterých jsou umisťovány přepravky s vychystanými položkami. Jakmile jsou prázdné přepravky, do kterých je umisťováno zboží při vychystávání,

k dispozici na začátku uliček regálového systému (tj. na čele regálů) či ve spádových regálech v dostatečném počtu, přerušuje operátor zodpovědný za vychystávání položek chůzi následujícími činnostmi:

1. fyzické umístění prázdných přepravek do označených lokací manipulačního vozíku, párování přepravky s lokací na vozíku pomocí skenování čárových kódů přepravky a lokace,
2. vybírání zboží na skladových lokacích a jeho umístění do přepravky na vozíku, skenování čárových kódů vybíraných položek, skladových lokací a lokací na vozíku, potvrzení odebraného množství,
3. umístění přepravek s vychystanými položkami do drah spádových regálů, párování přepravky s dráhou spádového regálu skenováním čárového kódu přepravky s čárovým kódem dráhy.

Veškeré činnosti spojené s vychystáváním položek musí efektivně řídit WMS systém. Efektivita v tomto případě znamená, že operátor logistiky může zamířit s vychystaným zbožím ke spádovým regálům, aniž by absolvoval chůzi oběma skladovými uličkami například v situaci, kdy k naplnění vozíku dojde již v polovině první uličky. Nebo operátor sice projde obě uličky, ale z první do druhé již projde průchodem, který je v regálovém systému umístěn uprostřed řady. Vzhledem k tomu, že nejobratkovější položky z hlediska frekvence, s jakou se objevují na objednávkách, jsou umístěny právě v první polovině regálového systému, bude k těmto případům docházet relativně často.

K ověření efektivity nového systému vychystávání položek byl využit obecný simulační model materiálových toků. Manipulace simulované pomocí ZMS představují činnosti popsané v bodech 1–3, k simulaci chůze mezi skladovými lokacemi a spádovými regály byl využit princip popsaný v kapitole 5.2.2. Předpokládaná rychlost chůze skladníka byla 4 km/h, trvání činností popsaných v bodě 1 bylo 5 s/přepravka a trvání činností popsaných v bodě 3 bylo 6 s/přepravka. Simulovaná intenzita materiálového toku odpovídala období špičky jak v počtu objednávek, tak v poptávaném množství (měsíc 10 na obrázku 5.36), přičemž ze simulace byly vyjmuty distribuční kanály 4 a 5 z důvodu možnosti výskytu příliš vysokých objednávaných množství a také kanály 2 a 7, u kterých jsou počty objednávek a poptávaná množství zanedbatelná. Počty položek na objednávkách byly generovány náhodně pomocí informací v tabulkách 5.1–5.4, metodou popsanou v kapitole 2.3.5. Náhodně byly generovány také lokace položek v regálovém systému s respektem k rozdělení skladových lokací pomocí Paretova pravidla na části A, B, C dle frekvence, s jakou se položky objevují na objednávkách.

Simulace byla provedena s cílem najít odpověď na otázku:

- O kolik % se zkrátí proces vychystávání položky oproti současnému stavu?

Vzhledem k tomu, že na délku procesu vychystávání položky v nově navrženém systému má zásadní vliv maximální počet přepravek, které operátor umístí na manipulační vozík, a maximální počet kusů všech položek, které jsou v jedné přepravce, bylo vytvořeno 10 různých scénářů, jež byly následně podrobeny simulaci. Tyto scénáře popisuje následující tabulka:

Tabulka 7.7 Simulované scénáře

Scénář	Maximální počet přepravek na vozíku	Maximální počet kusů všech položek v jedné přepravce
1	6	5
2	6	10
3	6	15
4	6	20
5	6	25
6	8	5
7	8	10
8	8	15
9	8	20
10	8	25

Maximální počet přepravek na vozíku je volen s ohledem na reálnou velikost manipulačního vozíku a reálné rozměry plastových přepravek. Maximální počet kusů všech položek v jedné přepravce je volen s ohledem na maximální váhu jednoho kusu položky (tj. 2 kg) a představuje maximální hmotnost, kterou bude operátor logistiky zvedat při umisťování přepravky s vychystaným zbožím do dráhy spádového regálu před stanovištěm kontroly a balení. Jde tedy o maximální hmotnost 50 kg. Kombinace maximálního počtu přepravek na vozíku a maximálního počtu kusů všech položek v jedné přepravce pak představuje maximální hmotnost, kterou bude operátor logistiky manipulovat na manipulačním vozíku. V tomto případě se jedná o hmotnost maximálně 400 kg.

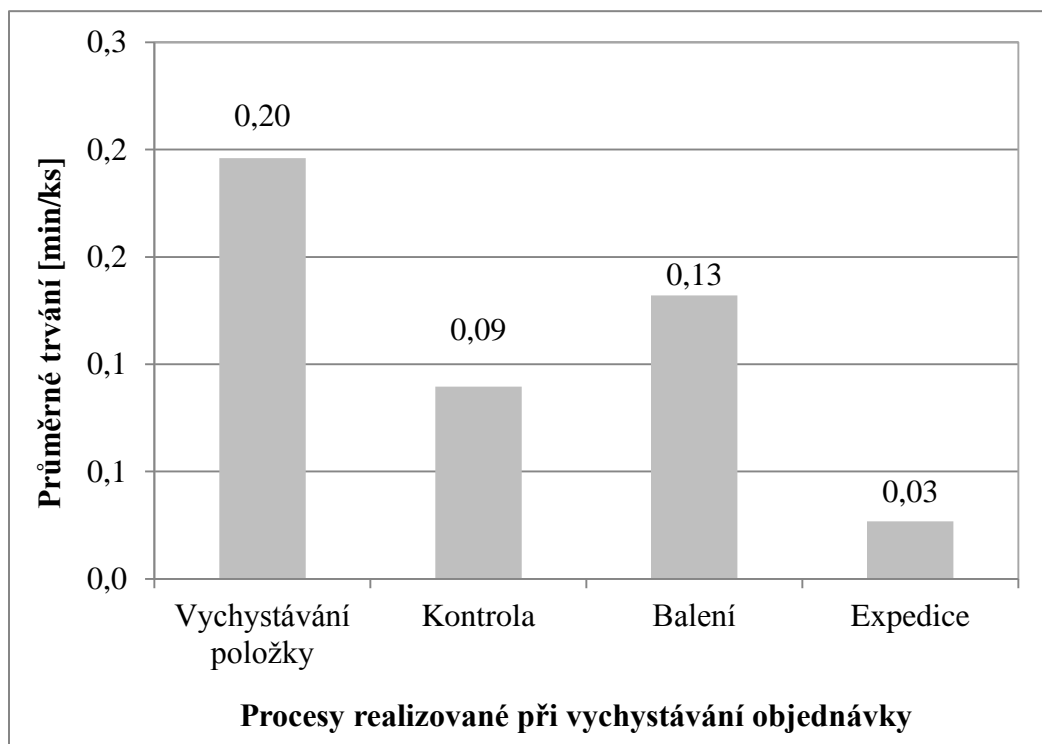
Simulace pokrývala období 30 dnů a pro každý simulovaný scénář bylo provedeno 1 000 opakování. Výstupy simulace byly zpracovány do podoby následující tabulky:

Tabulka 7.8 Výstupy simulace

Scénář	Maximální počet přepravek na vozíku	Maximální počet kusů všech položek v jedné přepravce	Průměrné zkrácení procesu vychystávání položky oproti současnému stavu [%]
1	6	5	prodloužení o 1 %
2	6	10	15 %
3	6	15	19 %

4	6	20	21 %
5	6	25	22 %
6	8	5	7 %
7	8	10	20 %
8	8	15	23 %
9	8	20	26 %
10	8	25	27 %

Uvedené výstupy pokrývají pouze objednávky distribučních kanálů 1, 3, a 6, a proto je nutné přepočítat průměrné zkrácení procesu vychystávání položky oproti současnému stavu také na objednávky ostatních kanálů. Za předpokladu, že by objednávky těchto kanálů byly zpracovávány dle současných pravidel, by měl výkonový profil v případě scénáře číslo 3 (tj. nejpravděpodobnější scénář z hlediska proveditelnosti) následující podobu:



Obr. 7.47 Výkonový profil vychystávání objednávky – návrh nového systému

Odhadované zkrácení procesu vychystávání položky oproti současnému stavu je v tomto případě cca 15%.

8 ZÁVĚR

Obecný simulační model materiálových toků je primárně určen k využití v úlohách zabývajících se alokací, návrhem, ověřením či optimalizací struktury skladovacích,

dopravních a výrobních kapacit v dodavatelských systémech. S tímto problémem také úzce souvisí stanovení potřeby zdrojů nutných k zajištění realizace materiálových toků o požadované intenzitě. Na druhou stranu není obecný simulační model materiálových toků vhodný k řešení úloh, které se zabývají operativním plánováním či výběrem dodavatelů. Simulační model umožňuje na jedné platformě propojení procesů probíhajících při realizaci materiálových toků a jejich optimalizaci s ohledem na nejrůznější strategie řízení a kritéria hodnocení výkonu. Struktura modelu je velmi jednoduchá a využívá pouze minimum předdefinovaných elementů, jež jsou k dispozici v prostředí simulačního softwaru Witness, ve kterém je model vytvořen. Tyto elementy jsou propojeny programovým kódem vytvořeným v jazyce VB do blokové struktury, která umožňuje simulaci materiálového toku na základě principu, že jakýkoli tok lze rozložit do konečného počtu pohybů. Uvedený princip umožňuje efektivní propojování procesů probíhajících ve výrobě, v dopravě, při skladování či distribuci a také alokaci zdrojů nutných k jejich realizaci. Kombinací obecného simulačního modelu materiálových toků s matematickými či heuristickými modely lze ověřovat různé strategie řízení výrobních, distribučních a skladovacích systémů dále i potřebu a využití zdrojů či například kvantifikovat efekty vyplývající z aplikací principů spolupráce.

Jednoduchost modelu a využití programovacího jazyka VB umožňuje přenést obecný model i do prostředí dalších softwarových produktů jako například Simul8 či Arena, tj. produktů, které disponují podobnými předdefinovanými elementy jako Witness a umožňují editaci programového kódu v jazyce, který je obdobou VB. Uvedené softwarové produkty navíc, stejně jako Witness, umožňují propojení s aplikací Microsoft Excel pro načítání vstupních dat a úpravu výstupů.

Velmi důležitou vlastností obecného modelu je možnost efektivně regulovat detail modelovaného systému pomocí počtu pohybů, do kterých lze rozložit materiálový tok, a také pomocí způsobu generování a přiřazování požadavků na provedení pohybu v simulaci. Generování požadavků na pohyb je prováděno vždy na začátku sekvence na sebe navazujících pohybů s cílem zahájit či účelně přerušit materiálový tok, který má být simulován. Používá se jednak ke generování požadavků na provedení pohybu, který stojí na začátku materiálového toku, a také ke generování požadavků v případě, že je z nějakého důvodu výhodnější nesimulovat všechny pohyby, do kterých lze tok rozložit. To je užitečné zejména v případě, že simulace některých pohybů je zbytečná z pohledu cíle, kterého má být dosaženo. Simulace těchto pohybů by pouze neúčelně zvyšovala rozsah modelu, a tím zpomalovala jeho běh. Způsob kódování požadavků na provedení pohybu do elementů typu Variable rovněž umožňuje efektivně zkracovat dobu běhu simulace, a to zejména pomocí volby množství, které je předmětem realizace pohybu. Uvedené vlastnosti modelu umožňují simulace rozsáhlých dodavatelských systémů na bázi historických vstupních dat pokrývajících dostatečně dlouhé období (obvykle 1–3 roky) v relativně krátkém čase (řádově jednotky minut na 1 běh simulace).

Ačkoli Witness nabízí uživatelům možnost zahrnout do simulačního modelu 2D či 3D grafiku, a to především z důvodu lepšího pochopení logiky modelovaného systému či k

názorné prezentaci dopadu změn na modelovaný systém, samotný obecný simulační model materiálových toků se v základní struktuře omezuje pouze na zobrazení využití zdrojů v průběhu simulace pomocí grafických elementů. To opět souvisí s potřebou efektivního běhu simulace pro účely výpočtů – simulace je spouštěna v režimu Batch, který s grafikou nepracuje. V případě požadavku na zobrazení materiálových toků například v layoutu výrobní haly či v distribučním systému se obvykle obecný model propojuje s modelem grafickým pomocí výstupů, které mají podobu záznamu pohybů tvořících materiálový tok. Grafický model může být vytvořen v prostředí Witness, často se ale také využívají softwarové produkty pro 3D simulace, jako je například VR Tara Builder.

I přesto, že se obecný simulační model materiálových toků jeví jako velmi výkonný nástroj vhodný k návrhu a optimalizaci struktury dodavatelských systémů, kvalita výstupů z tohoto modelu je silně závislá na několika faktorech. Nejdůležitějším z nich je ochota sdílet informace (například o poptávce, zásobách, dostupných výrobních a skladových kapacitách, procesech aj.) napříč dodavatelským systémem. Jedná se často o velmi obtížně řešitelný problém, vezmeme-li v úvahu, že jednotlivé články dodavatelského systému tvoří zpravidla různé obchodní organizace, které mají své vlastní cíle a prostředky pro jejich dosažení. Další úskalí představuje neochota obchodních organizací poskytovat informace, které vedou ke kvantifikaci efektů vyplývajících z případné optimalizace struktury dodavatelského systému. To má často za následek významné obtíže při hledání dohody, jakým způsobem dělit tyto efekty mezi jednotlivé prvky dodavatelského systému.

9 LITERATURA

Aalaei, A.; Davoudpour, H., A robust optimization model for cellular manufacturing system into supply chain management. *International Journal of Production Economics* **2017**, *183*, 667-679.

Aamodt, A.; Plaza, E., Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications* **1994**, *7* (1), 39-59.

Abdulwahab, U.; Wahab, M. I. M., Approximate dynamic programming modeling for a typical blood platelet bank. *Comput. Ind. Eng.* **2014**, *78*, 259-270.

Abogrean, E. M., Using Simulation to model of SCM Libyan cement factory. *2014 Science and Information Conference (Sai)* **2014**, 207-212.

Abu-Taieh, E.; El Sheikh, A. A. R., Commercial simulation packages: a comparative study. *International Journal of Simulation* **2007**, *8* (2), 66-76.

Agarwal, A.; Shankar, R.; Tiwari, M., Modeling the metrics of lean, agile and leagile supply chain: An ANP-based approach. *European Journal of Operational Research* **2006**, *173* (1), 211-225.

Agarwal, G.; Vijayvargy, L., Green Supplier Assessment in Environmentally Responsive Supply Chains through Analytical Network Process. In *International Multiconference of Engineers and Computer Scientist, Imecs 2012, Vol Ii*, Ao, S. I.; Castillo, O.; Douglas, C.; Feng, D. D.; Lee, J. A., Eds. Int Assoc Engineers-Iaeng: Hong Kong, 2012; 1218-1223.

Aguirre, A.; Muller, E.; Seffino, S.; Mendez, C. A.; Ieee, *Applying a simulation-based tool to productivity management in an automotive-parts industry*. Ieee: New York, 2008; 1838.

Aksin, Z.; Armony, M.; Mehrotra, V., The Modern Call Center: A Multi-Disciplinary Perspective on Operations Management Research. *Production and Operations Management* **2007**, *16* (6), 665-688.

Alfonso-Lizarazo, E. H.; Montoya-Torres, J. R.; Gutierrez-Franco, E., Modeling reverse logistics process in the agro-industrial sector: The case of the palm oil supply chain. *Appl. Math. Model.* **2013**, *37* (23), 9652-9664.

Almaraz, S. D.; Azzaro-Pantel, C.; Montastruc, L.; Pibouleau, L.; Senties, O. B., Assessment of mono and multi-objective optimization to design a hydrogen supply chain. *International Journal of Hydrogen Energy* **2013**, *38* (33), 14121-14145.

Ammeri, A.; Dammak, M. C.; Chabchoub, H.; Hachicha, W.; Masmoudi, F., *A Simulation Optimization Approach-based Genetic Algorithm for Lot Sizing Problem in a MTO Sector*. Ieee: New York, 2013; 476-481.

- Anderson Jr, E. G.; Fine, C. H.; Parker, G. G., Upstream volatility in the supply chain: The machine tool industry as a case study. *Production and Operations Management* **2000**, 9 (3), 239-261.
- Anzaldi, G.; Rubion, E.; Corchero, A., Efficient energy management in water distribution through applying case-based reasoning (cbr). *Proceedings of the 36th Iahr World Congress: Deltas of the Future and What Happens Upstream* **2015**, 5880-5888.
- Arasteh, A.; Aliahmadi, A.; Omran, M. M., Considering the business system's complexity with a network approach. *Int. J. Adv. Manuf. Technol.* **2014**, 70 (5-8), 869-885.
- Armenzoni, M.; Bottani, E.; Rinaldi, M.; Gallo, S. A.; Montanari, R., Analysis, Simulation and Optimization of the Milking Process in a Cowshed for the Production of Parmigiano Reggiano. *Int. J. Food Eng.* **2016**, 12 (9), 851-865.
- Ash, T., Dynamic node creation in backpropagation networks. *Connection science* **1989**, 1 (4), 365-375.
- Ashayeri, J.; Keij, R.; Bröker, A., Global business process re-engineering: A system dynamics-based approach. *International Journal of Operations and Production Management* **1998**, 18 (9-10), 817-831.
- Ashlaghi, M. J., A new approach to green supplier selection based on fuzzy multi-criteria decision making method and linear physical programming. *Teh. Vjesn.* **2014**, 21 (3), 591-597.
- Audy, J. F.; Damours, S.; Ronnqvist, M., An empirical study on coalition formation and cost/savings allocation. *International Journal of Production Economics* **2012**, 136 (1), 13-27.
- Ayag, Z., An integrated approach to evaluating conceptual design alternatives in a new product development environment. *International Journal of Production Research* **2005**, 43 (4), 687-713.
- Azadeh, A.; Raoofi, Z.; Zarrin, M., A multi-objective fuzzy linear programming model for optimization of natural gas supply chain through a greenhouse gas reduction approach. *J. Nat. Gas Sci. Eng.* **2015**, 26, 702-710.
- Azadeh, A.; Shafiee, F.; Yazdanparast, R.; Heydari, J.; Keshvarparast, A., Optimum Integrated Design of Crude Oil Supply Chain by a Unique Mixed Integer Nonlinear Programming Model. *Ind. Eng. Chem. Res.* **2017**, 56 (19), 5734-5746.
- Azougagh, Y.; Benhida, K.; Elfezazi, S., Contribution to the modelling and analysis of logistics system performance by Petri nets and simulation models: Application in a supply chain. In *2nd International Manufacturing Engineering Conference and 3rd Asia-Pacific Conference on Manufacturing Systems*, Hamedon, Z., Ed. Iop Publishing Ltd: Bristol, 2016; Vol. 114.

- Babazadeh, R.; Razmi, J.; Rabbani, M.; Pishvae, M. S., An integrated data envelopment analysis-mathematical programming approach to strategic biodiesel supply chain network design problem. *Journal of Cleaner Production* **2017**, *147*, 694-707.
- Banks, J., *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*, 1998. Wiley IEEE **1998**.
- Banomyong, R.; Supatn, N., Developing a supply chain performance tool for SMEs in Thailand. *Supply Chain Management* **2011**, *16* (1), 20-31.
- Barratt, M.; Oliveira, A., Exploring the experiences of collaborative planning initiatives. *International Journal of Physical Distribution & Logistics Management* **2001**, *31* (4), 266-289.
- Barton, J.; Dalley, D.; Patel, V., Life cycle assessment for waste management. *Waste management* **1996**, *16* (1), 35-50.
- Beamon, B. M., Supply chain design and analysis: Models and methods. *International journal of production economics* **1998**, *55* (3), 281-294.
- Becker, S.; Le Cun, Y. In *Improving the convergence of back-propagation learning with second order methods*, Proceedings of the 1988 connectionist models summer school, 1988; 29-37.
- Bellman, R. *The theory of dynamic programming*; Rand corp santa monica ca: 1954.
- Bevilacqua, M.; Ciarapica, F. E.; Mazzuto, G.; Postacchini, L.; Ieee, Air traffic management of an airport using discrete event simulation method. In *2012 Ieee International Conference on Industrial Engineering and Engineering Management*, Ieee: New York, 2012; 1034-1038.
- Biganzoli, E.; Boracchi, P.; Mariani, L.; Marubini, E., Feed forward neural networks for the analysis of censored survival data: a partial logistic regression approach. *Statistics in medicine* **1998**, *17* (10), 1169-1186.
- Blocher, J. D.; Garrett, R. W.; Schmenner, R. W., Throughput time reduction: Taking one's medicine. *Production and Operations Management* **1999**, *8* (4), 357-373.
- Boel, R.; Mihaylova, L., A compositional stochastic model for real time freeway traffic simulation. *Transportation Research Part B: Methodological* **2006**, *40* (4), 319-334.
- Bozarth, C. C.; Berry, W. L., Measuring the Congruence Between Market Requirements and Manufacturing: A Methodology and Illustration*. *Decision Sciences* **1997**, *28* (1), 121-150.
- Brailsford, S.; Hilton, N., A comparison of discrete event simulation and system dynamics for modelling health care systems. In Riley, J. (ed.) *Planning for the Future: Health Service*

Quality and Emergency Accessibility. Operational Research Applied to Health Services (ORAHs). Glasgow, 2001.

Brandenburg, M.; Govindan, K.; Sarkis, J.; Seuring, S., Quantitative models for sustainable supply chain management: Developments and directions. *European Journal of Operational Research* **2014**, *233* (2), 299-312.

Braun, M.; Golubitsky, M., *Differential equations and their applications*. Springer: 1983.

Braungart, M.; McDonough, W.; Bollinger, A., Cradle-to-cradle design: creating healthy emissions—a strategy for eco-effective product and system design. *Journal of cleaner production* **2007**, *15* (13), 1337-1348.

Burgess, K.; Singh, P. J., A proposed integrated framework for analysing supply chains. *Supply Chain Management* **2006**, *11* (4), 337-344.

Butler, J.; Morrice, D. J.; Mullarkey, P. W., A multiple attribute utility theory approach to ranking and selection. *Management Science* **2001**, *47* (6), 800-816.

Cadavid-Giraldo, N.; Velez-Gallego, M. C.; Guillen-Gosalbez, G., Technology Updating Decisions for Improving the Environmental Performance of an Operating Supply Chain: A Multiobjective Optimization Model for the Cement Industry. *Ind. Eng. Chem. Res.* **2016**, *55* (47), 12287-12300.

Cafaro, V. G.; Cafaro, D. C.; Mendez, C. A.; Cerda, J., oil-derivatives pipeline logistics using discrete-event simulation. In *Proceedings of the 2010 Winter Simulation Conference*, Johansson, B.; Jain, S.; MontoyaTorres, J.; Hukan, J.; Yucesan, E., Eds. Ieee: New York, 2010; 2101-2113.

Cachon, G.; Fisher, M., Campbell soup's continuous replenishment program: Evaluation and enhanced inventory decision rules. *Production and Operations Management* **1997**, *6* (3), 266-276.

Cambero, C.; Sowlati, T., Incorporating social benefits in multi-objective optimization of forest-based bioenergy and biofuel supply chains. *Appl. Energy* **2016**, *178*, 721-735.

Çetinkaya, S.; Lee, C. Y., Stock replenishment and shipment scheduling for vendor-managed inventory systems. *Management Science* **2000**, *46* (2), 217-232.

Cimino, A.; Longo, F.; Mirabelli, G., A general simulation framework for supply chain modeling: state of the art and case study. *arXiv preprint arXiv:1004.3271* **2010**.

Cinar, D.; Temur, G. T.; Topcu, Y. I., An Axiomatic Design Approach to the Classification of Reverse Logistics Network Design Studies Under Fuzziness. In *Supply Chain Management under Fuzziness: Recent Developments and Techniques*, Kahraman, C.; Oztaysi, B., Eds. Springer-Verlag Berlin: Berlin, 2014; Vol. 313, 639-654.

- Clarke, G. u.; Wright, J. W., Scheduling of vehicles from a central depot to a number of delivery points. *Operations research* **1964**, 12 (4), 568-581.
- Cook, W. D.; Seiford, L. M., Data envelopment analysis (DEA)–Thirty years on. *European Journal of Operational Research* **2009**, 192 (1), 1-17.
- Cooper, M. C.; Lambert, D. M.; Pagh, J. D., Supply chain management: more than a new name for logistics. *The international journal of logistics management* **1997**, 8 (1), 1-14.
- David, R.; Alla, H., *Discrete, Continuous, and Hybrid Petri Nets*. Springer Berlin Heidelberg: 2010.
- de Albuquerque, G. A.; Maciel, P.; Lima, R. M. F.; Magnani, F., Strategic and Tactical Evaluation of Conflicting Environment and Business Goals in Green Supply Chains. *Ieee Trans. Syst. Man Cybern. -Syst.* **2013**, 43 (5), 1013-1027.
- de Bruin, A. M.; van Rossum, A. C.; Visser, M. C.; Koole, G. M., Modeling the emergency cardiac in-patient flow: an application of queuing theory. *Health Care Management Science* **2007**, 10 (2), 125-137.
- Dlouhý, M.; Fábry, J.; Hladík, T.; Kuncová, M., Simulace podnikových procesů. Edn. 1. Brno. Computer Press: 2007.
- Dong, H.; Zhao, W. L.; Li, Y. P., A Novel Predictive Control Strategy for Supply Chain Management under Networked Manufacturing. In *Advanced Design Technology*, Jiang, Z. Y.; Han, J. T.; Liu, X. H., Eds. Trans Tech Publications Ltd: Durnten-Zurich, 2012; Vol. 421, 548.
- Dorigo, M.; Stutzle, T., The ant colony optimization metaheuristic: Algorithms, applications, and advances. *International series in operations research and management science* **2003**, 251-286.
- Dotoli, M.; Epicoco, N.; Falagario, M., A Technique for Supply Chain Network Design under Uncertainty using Cross-Efficiency Fuzzy Data Envelopment Analysis. *Ifac Papersonline* **2015**, 48 (3), 634-639.
- Dotoli, M.; Mangini, A. M.; Falagario, M.; Sciancalepore, F.; Ieee, A novel formulation of the DEA model for application to supplier selection. In *2010 Ieee Conference on Emerging Technologies and Factory Automation*, Ieee: New York, 2010.
- Downham, D.; Roberts, F., Multiplicative congruential pseudo-random number generators. *The Computer Journal* **1967**, 10 (1), 74-77.
- Doyle, J.; Green, R., Efficiency and cross-efficiency in DEA: Derivations, meanings and uses. *Journal of the operational research society* **1994**, 45 (5), 567-578.

- Doytchinov, B.; Lehoczky, J.; Shreve, S., Real-Time Queues in Heavy Traffic with Earliest-Deadline-First Queue Discipline. *The Annals of Applied Probability* **2001**, *11* (2), 332-378.
- Dror, M.; Trudeau, P., Stochastic vehicle routing with modified savings algorithm. *European Journal of Operational Research* **1986**, *23* (2), 228-235.
- Düntsche, I.; Gediga, G., Rough set dependency analysis in evaluation studies: An application in the study of repeated heart attacks. *Informatics Research Reports* **1995**, *10*, 25-30.
- Dyntar, J., *Design of material handling system using dynamic simulation and agent-based paradigm*. Int Business Information Management Assoc-Ibima: Norristown, 2014; 2000.
- Dyntar, J.; Gros, I., *Matematické modely pro manažerské rozhodování*. 2 ed.; VŠCHT Praha: Praha, 2015; 303.
- Dyntar, J.; Kemrova, E.; Skvor, J., Application of Discrete Event Simulation in Butadiene Rubber Supply Chain Optimization. *KGK-Kautsch. Gummi Kunstst.* **2012**, *65* (3), 32-36.
- Dyntar, J.; Strachotova, D., *Witness Dynamic Simulation for Supply Chain Design and Optimisation*. Int Business Information Management Assoc-Ibima: Norristown, 2015; 346.
- Dyson, R. G.; Allen, R.; Camanho, A. S.; Podinovski, V. V.; Sarrico, C. S.; Shale, E. A., Pitfalls and protocols in DEA. *European Journal of operational research* **2001**, *132* (2), 245-259.
- Dyson, R. G.; Thanassoulis, E., Reducing weight flexibility in data envelopment analysis. *Journal of the Operational Research Society* **1988**, *39* (6), 563-576.
- Entezaminia, A.; Heydari, M.; Rahmani, D., A multi-objective model for multi-product multi-site aggregate production planning in a green supply chain: Considering collection and recycling centers. *J. Manuf. Syst.* **2016**, *40*, 63-75.
- Ertugrul, I.; Ozbay, B., Supply chain optimization and distribution network application with AHP in a yarn company. *Tekst. Konfeksiyon* **2013**, *23* (2), 87-93.
- Evans, J. R.; Olson, D. L., *Introduction to simulation and risk analysis*. Prentice-Hall, Inc.: 1998.
- Farrell, M. J.; Fieldhouse, M., Estimating efficient production functions under increasing returns to scale. *Journal of the Royal Statistical Society. Series A (General)* **1962**, 252-267.
- Fishman, G. S., *A First Course in Monte Carlo*. Thomson Brooks/Cole: 2005.
- Fleisch, E.; Tellkamp, C., Inventory inaccuracy and supply chain performance: A simulation study of a retail supply chain. *International Journal of Production Economics* **2005**, *95* (3), 373-385.

Forrester, J. W., *The impact of feedback control concepts on the management sciences*. Foundation for Instrumentation Education and Research: 1960.

Forrester, J. W., Industrial dynamics: a major breakthrough for decision makers. *Harvard business review* **1958**, 36 (4), 37-66.

Fotr, J.; Plevný, M.; Svecová, L.; Vacík, E., Multi-criteria project portfolio optimization under risk and specific limitations. *E+ M Ekonomie a Management* **2013**, (4), 71.

Fousek, J.; Kuncova, M.; Fabry, J., *Discrete event simulation - production model in simul8*. European Council Modelling & Simulation: Nottingham, 2017; 229-234.

Fu, M. C., Optimization for simulation: Theory vs. practice. *INFORMS Journal on Computing* **2002**, 14 (3), 192-215.

Gandhi, K.; Govindan, K.; Jha, P. C., Fuzzy bi-criteria decision making approach for supplier selection and distribution network planning in supply chain management. *J. Inform. Optim. Science* **2016**, 37 (5), 653-679.

Gao, J. Y.; You, F. Q., Game theory approach to optimal design of shale gas supply chains with consideration of economics and life cycle greenhouse gas emissions. *Aiche J.* **2017**, 63 (7), 2671-2693.

Garcia-Caceres, R. G.; Martinez-Avella, M. E.; Palacios-Gomez, F., Tactical optimization of the oil palm agribusiness supply chain. *Appl. Math. Model.* **2015**, 39 (20), 6375-6395.

Ghosh, M.; Manna, S.; Chakrabarti, B., The travelling salesman problem on a dilute lattice: a simulated annealing study. *Journal of Physics A: Mathematical and General* **1988**, 21 (6), 1483.

Giaglis, G. M.; Mohamed Udin, Z.; Khan, M. K.; Zairi, M., A collaborative supply chain management framework: Part 1-planning stage. *Business process management journal* **2006**, 12 (3), 361-376.

Glasserman, P., *Monte Carlo methods in financial engineering*. Springer Science & Business Media: 2003; Vol. 53.

Glover, F.; Kelly, J. P.; Laguna, M. In *New advances for wedding optimization and simulation*, Simulation Conference Proceedings, 1999 Winter, IEEE: 1999; 255-260.

Glover, F.; Laguna, M., Tabu Search. In *Handbook of Combinatorial Optimization: Volume 1-3*, Du, D.-Z.; Pardalos, P. M., Eds. Springer US: Boston, MA, 1999; 2093-2229.

Gonela, V.; Zhang, J.; Osmani, A., Stochastic optimization of sustainable industrial symbiosis based hybrid generation bioethanol supply chains. *Comput. Ind. Eng.* **2015**, 87, 40-65.

Gong, D.; Liu, S.; Lu, X., Modelling the impacts of resource sharing on supply chain efficiency. *International Journal of Simulation Modelling* **2015**, 14 (4), 744-755.

- Gordon, G., *System simulation*. Prentice-Hall: Englewood Cliffs, N.J., 1978.
- Gottlich, S.; Kuhn, S., Routing strategies in production networks with random breakdowns. *Communications in Mathematical Sciences* **2016**, 14 (7), 1799-1820.
- Govindan, K.; Darbari, J. D.; Agarwal, V.; Jha, P. C., Fuzzy multi-objective approach for optimal selection of suppliers and transportation decisions in an eco-efficient closed loop supply chain network. *Journal of Cleaner Production* **2017**, 165, 1598-1619.
- Griffis, S. E.; Bell, J. E.; Closs, D. J., Metaheuristics in Logistics and Supply Chain Management. *Journal of Business Logistics* **2012**, 33 (2), 90-106.
- Grigoroudis, E.; Petridis, K.; Arabatzis, G., RDEA: A recursive DEA based algorithm for the optimal design of biomass supply chain networks. *Renew. Energy* **2014**, 71, 113-122.
- Gros, I. *Logistika. 1. vyd. Praha: VŠCHT, 1996. 228 s; ISBN 80-7080-262-6: 1996.*
- Gros, I., *Matematické modely pro manažerské rozhodování*. Vydavatelství VŠCHT: 2009.
- Gros, I.; Grosová, S., *Dodavatelské systémy: supply chain management*. Vysoká škola logistiky: 2012.
- Gros, I.; Grosová, S., Systémový přístup při navrhování dodavatelských systémů. *Acta Logistica Moravica* **2011**, 1, 43 - 50.
- Gros, I.; Grosová, S.; Dyntax, J., Importance of the system identification in supply systems modelling, theory and praxis. *Transport & Logistics* **2009**, (6), 75-79.
- Gueller, M.; Uygun, Y.; Noche, B., Simulation-based optimization for a capacitated multi-echelon production-inventory system. *Journal of Simulation* **2015**, 9 (4), 325-336.
- Guchhait, P.; Maiti, M. K.; Maiti, M., An EOQ model of deteriorating item in imprecise environment with dynamic deterioration and credit linked demand. *Appl. Math. Model.* **2015**, 39 (21), 6553-6567.
- Guide Jr, V. D. R.; Van Wassenhove, L. N., Or forum-the evolution of closed-loop supply chain research. *Operations research* **2009**, 57 (1), 10-18.
- Hailu, H.; Jilcha, K.; Birhan, E., Response Time Reduction in the Leather Products Manufacturing Industry Using Arena Simulation Method. In *Afro-European Conference for Industrial Advancement, Aecia 2014*, Abraham, A.; Kromer, P.; Snasel, V., Eds. Springer-Verlag Berlin: Berlin, 2015; Vol. 334, 271-281.
- Han, S. H.; Chu, C. H., Developing a collaborative supply chain reference model for a regional manufacturing industry in China. *International Journal of Electronic Customer Relationship Management* **2009**, 3 (1), 52-70.
- Haq, A. N.; Kannan, G., Fuzzy analytical hierarchy process for evaluating and selecting a vendor in a supply chain model. *The International Journal of Advanced Manufacturing Technology* **2006**, 29 (7-8), 826-835.

Harrison, A.; Van Hoek, R. I., *Logistics management and strategy: competing through the supply chain*. Pearson Education: 2008.

Harzheim, E., *Ordered sets*. Springer Science & Business Media: 2006; Vol. 7.

Hassini, E.; Surti, C.; Searcy, C., A literature review and a case study of sustainable supply chains with a focus on metrics. *International Journal of Production Economics* **2012**, *140* (1), 69-82.

Hebb, D. O., *The organization of behavior: A neuropsychological theory*. New York: Wiley: 1949.

Hecht-Nielsen, R., Theory of the backpropagation neural network. *Neural Networks* **1988**, *1* (Supplement-1), 445-448.

Hiremath, N. C.; Sahu, S.; Tiwari, M. K., Multi objective outbound logistics network design for a manufacturing supply chain. *Journal of Intelligent Manufacturing* **2013**, *24* (6), 1071-1084.

Hlupic, V.; Irani, Z.; Paul, R., Evaluation framework for simulation software. *The International Journal of Advanced Manufacturing Technology* **1999**, *15* (5), 366-382.

Holmes, R.; Parker, R., A vehicle scheduling procedure based upon savings and a solution perturbation scheme. *Operational Research Quarterly* **1976**, 83-92.

Hoogeweegen, M. R.; van Liere, D. W.; Vervest, P. H. M.; van der Meijden, L. H.; de Lepper, I., Strategizing for mass customization by playing the business networking game. *Decision Support Systems* **2006**, *42* (3), 1402-1412.

Hsieh, F. S.; Lin, J. B., A Multiagent Approach for Managing Collaborative Workflows in Supply Chains. *Proceedings of the 2014 Ieee 18th International Conference on Computer Supported Cooperative Work in Design (Cscwd)* **2014**, 71-76.

Huang, Y. S.; Qiu, Z. L.; Liu, Q. C.; Ieee, *Supply Chain Network Design Based on Fuzzy Neural Network and PSO*. Ieee: New York, 2008; 2189-2193.

Hubscher, R.; Glover, F., Applying tabu search with influential diversification to multiprocessor scheduling. *Computers & Operations Research* **1994**, *21* (8), 877-884.

Hwang, C.-L.; Masud, A. S. M., *Multiple objective decision making—methods and applications: a state-of-the-art survey*. Springer Science & Business Media: 2012; Vol. 164.

Hynek, J., *Genetické algoritmy a genetické programování*. Grada: 2008.

Chahal, K.; Eldabi, T., A multi-perspective comparison for selection between system dynamics and discrete event simulation. *International Journal of Business Information Systems* **2010**, *6* (1), 4-17.

- Charnes, A.; Cooper, W. W.; Rhodes, E., Measuring the efficiency of decision making units. *European journal of operational research* **1978**, 2 (6), 429-444.
- Che, Z. H.; Chang, Y. F., Integrated methodology for supplier selection: the case of a sphygmomanometer manufacturer in taiwan. *J. Bus. Econ. Manag.* **2016**, 17 (1), 17-34.
- Che, Z. H.; Chiang, T. A., Designing a collaborative supply-chain plan using the analytic hierarchy process and genetic algorithm with cycle-time estimation. *International Journal of Production Research* **2012**, 50 (16), 4426-4443.
- Chen, F.; Drezner, Z.; Ryan, J. K.; Simchi-Levi, D., *The bullwhip effect: Managerial insights on the impact of forecasting and information on variability in a supply chain*. Springer: 1999.
- Chopra, S.; Meindl, P., *Supply chain management. Strategy, planning & operation*. Springer: 2007.
- Choy, K. L.; Ho, G. T. S.; Lee, C. K. H.; Lam, H. Y.; Cheng, S. W. Y.; Siu, P. K. Y.; Pang, G. K. H.; Tang, V.; Lee, J. C. H.; Tsang, Y. P., A recursive operations strategy model for managing sustainable chemical product development and production. *International Journal of Production Economics* **2016**, 181, 262-272.
- Christopher, M., *Logistics and supply chain management: creating value-added networks*. Pearson education: 2005.
- Chung, C. A., *Simulation modeling handbook: a practical approach*. CRC press: 2003.
- Ilgin, M. A.; Gupta, S. M., Environmentally conscious manufacturing and product recovery (ECMPRO): a review of the state of the art. *Journal of environmental management* **2010**, 91 (3), 563-591.
- In, B. M.; Curran, M. A., *Life-cycle assessment: inventory guidelines and principles*. CRC Press: 1994.
- Iyer, A. V.; Bergen, M. E., Quick response in manufacturer-retailer channels. *Management Science* **1997**, 43 (4), 559-570.
- Jahangirian, M.; Eldabi, T.; Naseer, A.; Stergioulas, L. K.; Young, T., Simulation in manufacturing and business: A review. *European Journal of Operational Research* **2010**, 203 (1), 1-13.
- Jakhar, S. K., Performance evaluation and a flow allocation decision model for a sustainable supply chain of an apparel industry. *Journal of Cleaner Production* **2015**, 87, 391-413.
- Jensen, I. G.; Munster, M.; Pisinger, D., Optimizing the supply chain of biomass and biogas for a single plant considering mass and energy losses. *European Journal of Operational Research* **2017**, 262 (2), 744-758.

Jin, M. Z.; Ma, R. M.; Yao, L. F.; Ren, P. Y., An Effective Heuristic Algorithm for Robust Supply Chain Network Design under Uncertainty. *Appl. Math. Inf. Sci.* **2014**, 8 (2), 819-826.

Jindal, A.; Sangwan, K. S., Multi-objective fuzzy mathematical modelling of closed-loop supply chain considering economical and environmental factors. *Annals of Operations Research* **2017**, 257 (1-2), 95-120.

Jura, P., *Fuzzy logika v modelování a řízení dynamických systémů: současný stav, perspektivy a výuka*. Vutium: 2005.

Kadadevaramath, R. S.; Chen, J. C. H.; Shankar, B. L.; Rameshkumar, K., Application of particle swarm intelligence algorithms in supply chain network architecture optimization. *Expert Syst. Appl.* **2012**, 39 (11), 10160-10176.

Kalaitzidou, M. A.; Longinidis, P.; Georgiadis, M. C., Optimal design of closed-loop supply chain networks with multifunctional nodes. *Computers & Chemical Engineering* **2015**, 80, 73-91.

Kalwani, M. U.; Narayandas, N., Long-term manufacturer-supplier relationships: do they pay off for supplier firms? *The Journal of Marketing* **1995**, 1-16.

Kelton, W. D.; Law, A. M., *Simulation modeling and analysis*. McGraw Hill Boston: 2000.

Kennedy, J.; Eberhart, R. In *Particle swarm optimization*, Neural Networks, 1995. Proceedings., IEEE International Conference on, Nov/Dec 1995; 1995; 1942-1948 vol.4.

Kim, M.; Kim, J., Optimization model for the design and analysis of an integrated renewable hydrogen supply (IRHS) system: Application to Korea's hydrogen economy. *International Journal of Hydrogen Energy* **2016**, 41 (38), 16613-16626.

Knuth, D., Seminumerical Algorithms. Third edn. Volume 2 of The Art of Computer Programming. Addison Wesley: 1997.

Kočí, V., Metoda posuzování životního cyklu a chemický průmysl. *Chem. Listy* **2010**, 104, 921-925.

Kohonen, T., Self-organized formation of topologically correct feature maps. *Biological cybernetics* **1982**, 43 (1), 59-69.

Kolodner, J. L., Reconstructive memory: A computer model. *Cognitive science* **1983**, 7 (4), 281-328.

Kotzab, H., Improving supply chain performance by efficient consumer response? A critical comparison of existing ECR approaches. *Journal of Business & Industrial Marketing* **1999**, 14 (5/6), 364-377.

Koulamas, C.; Antony, S.; Jaen, R., A survey of simulated annealing applications to operations research problems. *Omega* **1994**, 22 (1), 41-56.

- Krippendorff, K., *Content analysis: An introduction to its methodology*. Sage: 2012.
- Kumar, R. S.; Kondapaneni, K.; Dixit, V.; Goswami, A.; Thakur, L. S.; Tiwari, M. K., Multi-objective modeling of production and pollution routing problem with time window: A self-learning particle swarm optimization approach. *Comput. Ind. Eng.* **2016**, 99, 29-40.
- Lagos, C.; Paredes, F.; Niklander, S.; Cabrera, E., Solving a Distribution Network Design Problem by Combining Ant Colony Systems and Lagrangian Relaxation. *Stud. Inform. Control* **2015**, 24 (3), 251-260.
- Latorre-Biel, J. I.; Jimenez-Macias, E.; Blanco-Fernandez, J.; Martinez-Camara, E.; Saenz-Diez, J. C.; Perez-Parte, M., Decision Support System, Based on the Paradigm of the Petri Nets, for the Design and Operation of a Dairy Plant. *Int. J. Food Eng.* **2015**, 11 (6), 767-776.
- Lebowitz, M., Memory-based parsing. *Artificial Intelligence* **1983**, 21 (4), 363-404.
- Lee, H. L.; Padmanabhan, V.; Whang, S., The bullwhip effect in supply chains¹. *Sloan management review* **1997**, 38 (3), 93-102.
- Lee, J. E.; Chung, K. Y.; Lee, K. D.; Gen, M., A multi-objective hybrid genetic algorithm to minimize the total cost and delivery tardiness in a reverse logistics. *Multimed. Tools Appl.* **2015**, 74 (20), 9067-9085.
- Lee, J. E.; Lee, K. D., Integrated forward and reverse logistics model: a case study in distilling and sale company in korea. *Int. J. Innov. Comp. Inf. Control* **2012**, 8 (7A), 4483-4495.
- Lee, J. K.; Lee, M. W.; Chi, S. D., DEVS/HLA-Based Modeling and Simulation for Intelligent Transportation Systems. *Simulation* **2003**, 79 (8), 423-439.
- Lee, Y. H.; Kwon, S. G., The hybrid planning algorithm for the distribution center operation using tabu search and decomposed optimization. *Expert Syst. Appl.* **2010**, 37 (4), 3094-3103.
- Lenstra, J. K., *Local search in combinatorial optimization*. Princeton University Press: 2003.
- Li, X. T.; Ma, S. J.; Hu, J. H., Multi-search differential evolution algorithm. *Appl. Intell.* **2017**, 47 (1), 231-256.
- Li, Z. H.; Liu, Y. K.; Yang, G. Q., A new probability model for insuring critical path problem with heuristic algorithm. *Neurocomputing* **2015**, 148, 129-135.
- Lieder, M.; Asif, F. M. A.; Rashid, A.; Mihelic, A.; Kotnik, S., Towards circular economy implementation in manufacturing systems using a multi-method simulation approach to link design and business strategy. *Int. J. Adv. Manuf. Technol.* **2017**, 93 (5-8), 1953-1970.

Lin, F. R.; Pai, Y. H., Using multi-agent simulation and learning to design new business processes. *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*. **2000**, *30* (3), 380-384.

Liu, C. X.; Shu, T.; Chen, S.; Wang, S. Y.; Lai, K. K.; Gan, L., An improved grey neural network model for predicting transportation disruptions. *Expert Syst. Appl.* **2016**, *45*, 331-340.

Liu, L. J.; Xu, W. J., *The Researches on the Petri Net Based Optimization and Empirical Study on Customized Enterprises Supply Chain*. China Machine Press: Beijing, 2008; 412-416.

Liu, M. Z.; Zhou, M. H.; Lv, X. Q.; Yang, J. X., The research on modeling of coal supply chain based on object-oriented Petri net and optimization. In *Proceedings of the International Conference on Mining Science & Technology*, Ge, S.; Liu, J.; Guo, C., Eds. Elsevier Science Bv: Amsterdam, 2009; Vol. 1, 1608-1616.

Lyneis, J. M., System dynamics for market forecasting and structural analysis. *System Dynamics Review* **2000**, *16* (1), 3-25.

Ma, Y. F.; Yan, F.; Kang, K.; Wei, X. G., A novel integrated production-distribution planning model with conflict and coordination in a supply chain network. *Knowledge-Based Syst.* **2016**, *105*, 119-133.

Macal, C. M.; North, M. J., Tutorial on agent-based modelling and simulation. *Journal of simulation* **2010**, *4* (3), 151-162.

Maiyar, L. M.; Thakkar, J. J., A combined tactical and operational deterministic food grain transportation model: Particle swarm based optimization approach. *Comput. Ind. Eng.* **2017**, *110*, 30-42.

Malindžák, D., *Teória logistiky*. Karnat: Košice, 2007.

Manlig, F., Počítačová simulace diskrétních událostí. *MM-Průmyslové spektrum, r* **1999**, *3*, 34.

Marsaglia, G.; Zaman, A.; Tsang, W. W., Toward a universal random number generator. *Statistics & Probability Letters* **1990**, *9* (1), 35-39.

Mayring, P.; Gläser-Zikuda, M., *Die Praxis der Qualitativen Inhaltsanalyse*. Beltz Weinheim: 2008.

McCulloch, W. S.; Pitts, W., A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics* **1943**, *5* (4), 115-133.

McManus, M. D. M. P. H. Michael L.; Long, M. D. Michael C.; Cooper, A.; Litvak, P. D. E., Queuing Theory Accurately Models the Need for Critical Care Resources. *Anesthesiology* **2004**, *100* (5), 1271-1276.

- Mendes, A. R.; Ramos, A. L.; Simaria, A. S.; Vilarinho, P. M., Combining heuristic procedures and simulation models for balancing a PC camera assembly line. *Computers and Industrial Engineering* **2005**, 49 (3), 413-431.
- Mentzer, J. T.; DeWitt, W.; Keebler, J. S.; Min, S.; Nix, N. W.; Smith, C. D.; Zacharia, Z. G., Defining supply chain management. *Journal of Business logistics* **2001**, 22 (2), 1-25.
- Mentzer, J. T.; Foggin, J. H.; Golicic, S. L., Collaboration: the enablers, impediments, and benefits. *Supply chain management review* **2000**, 4 (4), 52-58.
- Mirmajlesi, S. R.; Shafaei, R., An integrated approach to solve a robust forward/reverse supply chain for short lifetime products. *Comput. Ind. Eng.* **2016**, 97, 222-239.
- Mitchell, M., *An Introduction to Genetic Algorithms*. Bradford Books: 1998.
- Modrlák, O., Teorie automatického řízení II. *Fuzzy řízení a regulace. Praha, Technická universita v Liberci* **2004**.
- Moghaddam, M.; Nof, S. Y., Combined demand and capacity sharing with best matching decisions in enterprise collaboration. *International Journal of Production Economics* **2014**, 148, 93-109.
- Moore, W. J., *Fyzikální chemie*. SNTL: Praha, 1981.
- Mousazadeh, M.; Torabi, S. A.; Zahiri, B., A robust possibilistic programming approach for pharmaceutical supply chain network design. *Computers & Chemical Engineering* **2015**, 82, 115-128.
- Muresan, M.; Cormos, C. C.; Agachi, P. S., Biomass gasification-based hydrogen production supply chain analysis under demand variability. *Stud. Univ. Babes-Bolyai Chem.* **2014**, 59 (3), 29-42.
- Nasiri, G. R.; Zolfaghari, R.; Davoudpour, H., An integrated supply chain production-distribution planning with stochastic demands. *Comput. Ind. Eng.* **2014**, 77, 35-45.
- Orady, E. A.; Osman, T. A.; Bailo, C. P., Virtual reality software for robotics and manufacturing cell simulation. *Computers and Industrial Engineering* **1997**, 33 (1-2), 87-90.
- Osman, I. H.; Kelly, J. P., *Meta-heuristics: theory and applications*. Springer Science & Business Media: 2012.
- Owens, S. F.; Levary, R. R., Evaluating design alternatives of an extruded food production line using simulation. *Simulation* **2002**, 78 (10), 626-632.
- Ozdemir, A.; Cho, B. R., A Nonlinear Integer Programming Approach to Solving the Robust Parameter Design Optimization Problem. *Quality and Reliability Engineering International* **2016**, 32 (8), 2859-2870.

- Ozgen, D.; Gulsun, B., Combining possibilistic linear programming and fuzzy AHP for solving the multi-objective capacitated multi-facility location problem. *Inf. Sci.* **2014**, *268*, 185-201.
- Paessens, H., The savings algorithm for the vehicle routing problem. *European Journal of Operational Research* **1988**, *34* (3), 336-344.
- Pan, Y. C., *Simulation and Optimization for Supply Chain Based on Multi-agent Reinforcement Learning: A Case Study on a Large-scale Refinery*. Orient Acad Forum: Marrickville, 2008; 433-439.
- Pareto, V.; Page, A. N., Translation of Manuale di economia politica ("Manual of political economy"). *AM Kelley* **1971**.
- Park, S. K.; Miller, K. W., Random number generators: good ones are hard to find. *Communications of the ACM* **1988**, *31* (10), 1192-1201.
- Pawlak, Z., Rough sets. *International Journal of Parallel Programming* **1982**, *11* (5), 341-356.
- Pawlak, Z., Rough sets and fuzzy sets. *Fuzzy sets and Systems* **1985**, *17* (1), 99-102.
- Pawlak, Z.; Wong, S. K. M.; Ziarko, W., Rough sets: probabilistic versus deterministic approach. *International Journal of Man-Machine Studies* **1988**, *29* (1), 81-95.
- Payne, W.; Rabung, J. R.; Bogyo, T., Coding the Lehmer pseudo-random number generator. *Communications of the ACM* **1969**, *12* (2), 85-86.
- Pepino, A.; Torri, A.; Mazzitelli, A.; Tamburis, O.; Ieee, A simulation model for analyzing the nurse workload in a university hospital ward. In *2015 Winter Simulation Conference*, Ieee: New York, 2015; 1367-1378.
- Pernica, P., *Logistika pro 21. století. Supply Chain Management*. Radix: 2005.
- Persson, F.; Araldi, M., The development of a dynamic supply chain analysis tool-Integration of SCOR and discrete event simulation. *International Journal of Production Economics* **2009**, *121* (2), 574-583.
- Petri, C. A., *Kommunikation mit Automaten*. Rhein.-Westfäl. Inst. f. Instrumentelle Mathematik an der Univ. Bonn: 1962.
- Petridis, K.; Dey, P. K.; Emrouznejad, A., A branch and efficiency algorithm for the optimal design of supply chain networks. *Annals of Operations Research* **2017**, *253* (1), 545-571.
- Pfahl, D.; Klemm, M.; Ruhe, G., A CBT module with integrated simulation component for software project management education and training. *Journal of Systems and Software* **2001**, *59* (3), 283-298.

Pradenas, L.; Garces, J.; Parada, V.; Ferland, J., Genotype-phenotype heuristic approaches for a cutting stock problem with circular patterns. *Eng. Appl. Artif. Intell.* **2013**, 26 (10), 2349-2355.

Princ, P.; Cernohous, R.; Kuncova, M., Simulation Model of the Logistic Flows between Warehouse and Production. *Proceedings of the 29th International Conference on Mathematical Methods in Economics 2011, Pts I and II* **2011**, 581-586.

Rabelo, L.; Helal, M.; Jones, A.; Min, H.-S., Enterprise simulation: a hybrid system approach. *International Journal of Computer Integrated Manufacturing* **2005**, 18 (6), 498-508.

Remtová, K., *Posuzování životního cyklu-metoda LCA*. Ministerstvo životního prostředí: 2003.

Röder, A.; Tibken, B., A methodology for modeling inter-company supply chains and for evaluating a method of integrated product and process documentation. *European Journal of Operational Research* **2006**, 169 (3), 1010-1029.

Rodrigues, A. G.; Williams, T. M., System dynamics in project management: Assessing the impacts of client behaviour on project performance. *Journal of the Operational Research Society* **1998**, 49 (1), 2-15.

Rosenblatt, F., The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review* **1958**, 65 (6), 386.

Roser, C.; Nakano, M.; Tanaka, M., Single simulation buffer optimization. *JSME International Journal, Series C: Mechanical Systems, Machine Elements and Manufacturing* **2006**, 48 (4), 763-768.

Saaty, T. L., *Decision making with dependence and feedback: The analytic network process*. RWS publications Pittsburgh: 1996; Vol. 4922.

Saaty, T. L., Relative measurement and its generalization in decision making why pairwise comparisons are central in mathematics for the measurement of intangible factors the analytic hierarchy/network process. *Revista de la Real Academia de Ciencias Exactas, Fisicas y Naturales. Serie A. Matematicas* **2008**, 102 (2), 251-318.

Saaty, T. L., *Theory and applications of the analytic network process: decision making with benefits, opportunities, costs, and risks*. RWS publications: 2005.

Saaty, T. L., What is the Analytic Hierarchy Process? In *Mathematical Models for Decision Support*, Mitra, G.; Greenberg, H. J.; Lootsma, F. A.; Rijkaert, M. J.; Zimmermann, H. J., Eds. Springer Berlin Heidelberg: Berlin, Heidelberg, 1988; 109-121.

Sadeghi, J.; Mousavi, S. M.; Niaki, S. T. A.; Sadeghi, S., Optimizing a bi-objective inventory model of a three-echelon supply chain using a tuned hybrid bat algorithm. *Transp. Res. Pt. e-Logist. Transp. Rev.* **2014**, 70, 274-292.

- Sadeghi, J.; Taghizadeh, M.; Sadeghi, A.; Jahangard, R.; Tavakkoli-Moghaddam, R., Optimizing a vendor managed inventory (VMI) model considering delivering cost in a three-echelon supply chain using two tuned-parameter meta-heuristics. *Int. J. Syst. Assur. Eng. Manag.* **2015**, *6* (4), 500-510.
- Seuring, S.; Müller, M., From a literature review to a conceptual framework for sustainable supply chain management. *Journal of cleaner production* **2008**, *16* (15), 1699-1710.
- Shabani, N.; Sowlati, T.; Ouhimmou, M.; Ronnqvist, M., Tactical supply chain planning for a forest biomass power plant under supply uncertainty. *Energy* **2014**, *78*, 346-355.
- Shin, Y.; Moon, D., M/M/c Retrial Queue with Multiclass of Customers. *Methodology and Computing in Applied Probability* **2014**, *16* (4), 931-949.
- Schaffer, J. D.; Caruana, R. A.; Eshelman, L. J.; Das, R., *A study of control parameters affecting online performance of genetic algorithms for function optimization*. 1989; 51-60.
- Schank, R. C., *Dynamic memory: A theory of reminding and learning in computers and people*. cambridge university press: 1983.
- Schwaninger, M.; Janovjak, M.; Ambroz, K., Second-order intervention: Enhancing organizational competence and performance. *Systems Research and Behavioral Science* **2006**, *23* (4), 529-545.
- Simatupang, T. M.; Sridharan, R., A benchmarking scheme for supply chain collaboration. *Benchmarking: An International Journal* **2004**, *11* (1), 9-30.
- Soleimani, F.; Khamseh, A. A.; Naderi, B., Optimal decisions in a dual-channel supply chain under simultaneous demand and production cost disruptions. *Annals of Operations Research* **2016**, *243* (1-2), 301-321.
- Soleimani, H.; Govindan, K.; Saghafi, H.; Jafari, H., Fuzzy multi-objective sustainable and green closed-loop supply chain network design. *Comput. Ind. Eng.* **2017**, *109*, 191-203.
- Spedding, T. A.; Sun, G. Q., Application of discrete event simulation to the activity based costing of manufacturing systems. *International Journal of Production Economics* **1999**, *58* (3), 289-301.
- Spengler, T.; Schröter, M., Strategic Management of Spare Parts in Closed-Loop Supply Chains - A System Dynamics Approach. *Interfaces* **2003**, *33* (6), 7-17.
- Stank, T. P.; Keller, S. B.; Daugherty, P. J., Supply chain collaboration and logistical service performance. *Journal of Business logistics* **2001**, *22* (1), 29-48.
- Stock, J. R.; Boyer, S. L., Developing a consensus definition of supply chain management: a qualitative study. *International Journal of Physical Distribution & Logistics Management* **2009**, *39* (8), 690-711.

- Strachotova, D.; Pavlistik, J., *The Assessment of Efficiency of In-Plant Milk-Run Distribution System in Cable Manufacturing for Automotive Industry*. Int Business Information Management Assoc-Ibima: Norristown, 2017; 105-109.
- Swaminathan, J. M.; Smith, S. F.; Sadeh, N. M., Modeling supply chain dynamics: A multiagent approach*. *Decision sciences* **1998**, 29 (3), 607-632.
- Šíma, J.; Neruda, R. *Teoretické otázky neuronových sítí. 1. vydání. Praha: MATFYZPRESS, MFF UK*; ISBN 80-85863-18-9: 1996.
- Škvor, J.; Dyntar, J.; Gros, I., Efektivní design dodavatelských systémů. *Scientific Papers of the University of Pardubice* **2011**, (20), 167-182.
- Takagi, H., Unified and refined analysis of the response time and waiting time in the m/m/m fcfs preemptive-resume priority queue. *Journal of Industrial and Management Optimization* **2017**, 13 (4), 1945-1973.
- Tang, C. S.; Zhou, S., Research advances in environmentally and socially sustainable operations. *European Journal of Operational Research* **2012**, 223 (3), 585-594.
- Thunberg, M.; Persson, F. In *A logistics framework for improving construction supply chain performance*, Procs 29th Annual ARCOM Conference, Association of Researchers in Construction Management (ARCOM): 2013.
- Tobail, A.; Crowe, J.; Arisha, A., Serious gaming learning: supply chain multi-agent web-based simulation game. *5th International Conference of Education, Research and Innovation (Iceri 2012)* **2012**, 4120-4128.
- Tsao, Y. C.; Liao, L. W.; Sun, P. C., Dynamic decision-making for an inventory system with time-varying demand. *Math. Comput. Model. Dyn. Syst.* **2013**, 19 (5), 401-416.
- UmaDevi, K.; Elango, C.; Rajesh, R., International Conference on Modeling Optimization and Computing Vendor selection using AHP. In *International Conference on Modelling Optimization and Computing*, Rajesh, R.; Ganesh, K.; Koh, S. C. L., Eds. Elsevier Science Bv: Amsterdam, 2012; Vol. 38, 1946-1949.
- Validi, S.; Bhattacharya, A.; Byrne, P. J., A case analysis of a sustainable food supply chain distribution system-A multi-objective approach. *International Journal of Production Economics* **2014**, 152, 71-87.
- Validi, S.; Bhattacharya, A.; Byrne, P. J., A solution method for a two-layer sustainable supply chain distribution model. *Computers & Operations Research* **2015**, 54, 204-217.
- Van Laarhoven, P. J.; Aarts, E. H., *Simulated annealing: theory and applications*. Springer Science & Business Media: 1987; Vol. 37.

Vaskan, P.; Guillen-Gosalbez, G.; Turkay, M.; Jimenez, L., Multiobjective Optimization of Utility Plants under Several Environmental Indicators Using an MILP-Based Dimensionality Reduction Approach. *Ind. Eng. Chem. Res.* **2014**, *53* (50), 19559-19572.

Vávrová, V., *Řízení výroby a nákupu*. Grada: 2007.

Viergutz, C.; Knust, S., Integrated production and distribution scheduling with lifespan constraints. *Annals of Operations Research* **2014**, *213* (1), 293-318.

Wagner, B. A.; Macbeth, D. K.; Boddy, D., Improving supply chain relations: an empirical case study. *Supply Chain Management: An International Journal* **2002**, *7* (4), 253-264.

Wang, K. J.; Nguyen, P. H., Capacity planning with technology replacement by stochastic dynamic programming. *European Journal of Operational Research* **2017**, *260* (2), 739-750.

Wang, M.; Sun, G.; Wang, D. In *Manufacturing Simulation: An Effective Tool for Productivity Improvement*, Proceedings of 3rd International Microelectronics & Systems' 93 Conference, 1993.

Weiss, G., *Multiagent systems: a modern approach to distributed artificial intelligence*. MIT press: 1999.

Wenzler, I., Development of an asset management strategy for a network utility company: Lessons from a dynamic business simulation approach. *Simulation and Gaming* **2005**, *36* (1), 75-90.

Weston Jr, F. C.; Herrmann, F.; Davidoff, P. H., Capacity planning and process analysis a simulation study of a microbrewery. *Production and Inventory Management Journal* **1999**, *40* (2), 48-52.

Widrow, B., *Adaptive "adaline" Neuron Using Chemical "memistors"*. 1960.

Wiecek, P.; Altman, E.; Ghosh, A., Mean-Field Game Approach to Admission Control of an M/M/infinity Queue with Shared Service Cost. *Dynamic Games and Applications* **2016**, *6* (4), 538-566.

Williams, P.; Tai, G. F.; Lei, Y. M., Simulation based analysis of patient arrival to health care systems and evaluation of an operations improvement scheme. *Annals of Operations Research* **2010**, *178* (1), 263-279.

Winston, W. L.; Goldberg, J. B., *Operations research: applications and algorithms*. Duxbury press Boston: 2004; Vol. 3.

Wolstenholme, E. F., The use of system dynamics as a tool for intermediate level technology evaluation: Three case studies. *Journal of Engineering and Technology Management - JET-M* **2003**, *20* (3), 193-204.

Xiao, R. B.; Yu, T. Y.; Gong, X. G., Modeling and simulation of ant colony's labor division with constraints for task allocation of resilient supply chains. *Int. J. Artif. Intell. Tools* **2012**, 21 (3), 19.

Xu, D.; Nageshwaranier, S. S.; Son, Y. J.; Song, S. G., Simulation-based assessment of change propagation effect in an aircraft design process. In *Proceedings of the 2011 Winter Simulation Conference*, Jain, S.; Creasey, R.; Himmelspace, J., Eds. Ieee: New York, 2011; 1710-1721.

Yang, C. H.; Liu, H. B.; Ji, P.; Ma, X., Optimal acquisition and remanufacturing policies for multi-product remanufacturing systems. *Journal of Cleaner Production* **2016**, 135, 1571-1579.

Yang, M. S.; Li, Q.; Center for Supply Chain, M.; Logistics, The Optimization Research of Multi-level Logistic and Distribution Based on Dynamic Programming. In *Icoscm 2007 - International Conference on Operations and Supply Chain Management in China*, 2007; Vol. 1.

Yao, Y.; Evers, P. T.; Dresner, M. E., Supply chain integration in vendor-managed inventory. *Decision support systems* **2007**, 43 (2), 663-674.

Yin, S.; Nishi, T.; Grossmann, I. E., Optimal quantity discount coordination for supply chain optimization with one manufacturer and multiple suppliers under demand uncertainty. *Int. J. Adv. Manuf. Technol.* **2015**, 76 (5-8), 1173-1184.

Yousefi-Babadi, A.; Tavakkoli-Moghaddam, R.; Bozorgi-Amiri, A.; Seifi, S., Designing a Reliable Multi-Objective Queuing Model of a Petrochemical Supply Chain Network under Uncertainty: A Case Study. *Computers & Chemical Engineering* **2017**, 100, 177-197.

Yu, H.; Solvang, W. D., A general reverse logistics network design model for product reuse and recycling with environmental considerations. *Int. J. Adv. Manuf. Technol.* **2016**, 87 (9-12), 2693-2711.

Zadeh, L. A., Fuzzy sets. In *Fuzzy Sets, Fuzzy Logic, And Fuzzy Systems: Selected Papers by Lotfi A Zadeh*, World Scientific: 1996; 394-432.

Zamarripa, M. A.; Aguirre, A. M.; Mendez, C. A.; Espuna, A., Improving supply chain planning in a competitive environment. *Computers & Chemical Engineering* **2012**, 42, 178-188.

Zenios, S. A.; Holmer, M. R.; McKendall, R.; Vassiadou-Zeniou, C., Dynamic models for fixed-income portfolio management under uncertainty. *Journal of Economic Dynamics and Control* **1998**, 22 (10), 1517-1541.

Zha, X. F.; Lim, S. Y. E., Assembly/disassembly task planning and simulation using expert Petri nets. *International Journal of Production Research* **2000**, 38 (15), 3639-3676.

- Zhan, Z. H.; Zhang, J.; Li, Y.; Chung, H. S. H., Adaptive Particle Swarm Optimization. *IEEE Trans. Syst. Man Cybern. Part B-Cybern.* **2009**, *39* (6), 1362-1381.
- Zhang, H. R.; Liang, Y. T.; Liao, Q.; Wu, M. Y.; Yan, X. H., A hybrid computational approach for detailed scheduling of products in a pipeline with multiple pump stations. *Energy* **2017**, *119*, 612-628.
- Zhang, W. H.; Reimann, M., Towards a multi-objective performance assessment and optimization model of a two-echelon supply chain using SCOR metrics. *Cent. Europ. J. Oper. Res.* **2014**, *22* (4), 591-622.
- Zhang, Y.; Yang, A.; Xiong, C.; Wang, T.; Zhang, Z., Feature selection using data envelopment analysis. *Knowledge-Based Syst.* **2014**, *64*, 70-80.
- Zhang, Y. N.; Wright, M. M., Product Selection and Supply Chain Optimization for Fast Pyrolysis and Biorefinery System. *Ind. Eng. Chem. Res.* **2014**, *53* (51), 19987-19999.
- Zohal, M.; Soleimani, H., Developing an ant colony approach for green closed-loop supply chain network design: a case study in gold industry. *Journal of Cleaner Production* **2016**, *133*, 314-337.

SEZNAM OBRÁZKŮ

Obr. 4.1 Kvantitativní přístupy používané při modelování dodavatelských systémů	13
Obr. 4.2 Etapy tvorby simulačního modelu.....	54
Obr. 4.3 Uspořádání základních komponent v DES.....	61
Obr. 4.4 Diagram kauzálních smyček s pozitivní závislostí proměnných.....	63
Obr. 4.5 Diagram kauzálních smyček s rovnovážnou smyčkou a zpožděním	63
Obr. 4.6 Podíl využití modelových přístupů na celkovém počtu relevantních prací zahrnutých do literární rešerše – literární rešerše Beamon (1998).....	68
Obr. 4.7 Podíl využití modelových přístupů na celkovém počtu relevantních prací zahrnutých do literární rešerše – literární rešerše Ilgin and Gupta (2010)	68
Obr. 4.8 Podíl využití modelových přístupů na celkovém počtu relevantních prací zahrnutých do literární rešerše – literární rešerše Brandenburg, et al. (2014).....	69
Obr. 4.9 Počet publikací o modelování dodavatelských systémů se simulační metodou – literární rešerše Ilgin and Gupta (2010).....	69
Obr. 4.10 Počet publikací o modelování dodavatelských systémů se simulační metodou – literární rešerše Jahangirian, et al. (2010).....	70
Obr. 4.11 Podíl využití druhů simulace na celkovém počtu relevantních prací zahrnutých do literární rešerše – literární rešerše Ilgin and Gupta (2010)	70
Obr. 4.12 Podíl využití druhů simulace na celkovém počtu relevantních prací zahrnutých do literární rešerše – literární rešerše Jahangirian, et al. (2010).....	71
Obr. 5.1 Roztřídění softwarových produktů dle druhu simulace	73
Obr. 5.2 Aplikace softwarových produktů v odvětvích popsané ve vědecké literatuře	74
Obr. 5.3 Detail elementu Machine po jeho definování v modelu.....	80
Obr. 5.4 Posloupnost akcí a pravidel, kterými prochází entita v elementu Machine	81
Obr. 6.1 Obecný simulační model materiálových toků vytvořený v prostředí Witness.....	88
Obr. 6.2 Vstupní data simulace v prostředí MS Excel	95
Obr. 6.3 Detail elementu mPohybZdroj	97
Obr. 6.4 Detail elementu mManipulaceBod1	100
Obr. 6.5 Detail elementu mPohybBod1Bod2.....	102
Obr. 6.6 Detail elementu mManipulaceBod2.....	104
Obr. 6.7 Detail elementu mGeneratorPozPohyb1	107
Obr. 6.8 Detail elementu mCML.....	109
Obr. 6.9 Detail elementu pcZdroj1 pro grafické zobrazení využití zdrojů.....	115
Obr. 6.10 Detail element pcZdroj2 pro grafické zobrazení využití zdrojů.....	115

Obr. 6.11 Detail elementů tZdroj3 pro grafické zobrazení využití zdrojů	116
Obr. 6.12 Detail elementu mZapisVyuzitiZdroj.....	117
Obr. 6.13 Výstupy simulace v prostředí MS Excel	118
Obr. 7.1 Současná struktura distribučního systému	119
Obr. 7.2 Rozložení poptávky a její hustota	120
Obr. 7.3 Provozní náklady skladů	121
Obr. 7.4 Ilustrace transformace GPS souřadnic do roviny	123
Obr. 7.5 Srovnání skutečných nákladů na distribuci současného uspořádání distribučního systému se simulací	132
Obr. 7.6 Umístění skladu a návrh teritorií – varianta s 1 skladem	133
Obr. 7.7 Umístění skladů a návrh teritorií – varianta se 2 sklady	134
Obr. 7.8 Umístění skladů a návrh teritorií – varianta se 3 sklady	134
Obr. 7.9 Umístění skladů a návrh teritorií – varianta se 4 sklady	135
Obr. 7.10 Umístění skladů a návrh teritorií – varianta s 5 sklady	135
Obr. 7.11 Umístění skladů a návrh teritorií – varianta se 6 sklady	136
Obr. 7.12 Dopravní náklady C_1 a C_2	136
Obr. 7.13 Provozní náklady skladů C_3 a náklady z vázanosti kapitálu v zásobách C_4	137
Obr. 7.14 Celkové náklady na distribuci C	137
Obr. 7.15 Porovnání skutečných nákladů na distribuci s náklady na distribuci získanými simulací pro 6 různých variant	138
Obr. 7.16 Skladový areál společnosti	139
Obr. 7.17 Typický průběh fyzické kompletace objednávky zákazníka.....	140
Obr. 7.18 Využití skladníků společnosti	141
Obr. 7.19 Využití manipulační techniky používané skladníky	141
Obr. 7.20 Úpravy layoutu skladového areálu pro koncept řízeného skladování.....	142
Obr. 7.21 Struktura ZMS pro simulaci pohybu kompletace objednávky ve skladu stavební chemie a skladu maloobrátkového zboží.....	144
Obr. 7.22 Využití zakladačů pro VNA v prvním roce simulace	145
Obr. 7.23 Využití čelních vozíků v prvním roce simulace	145
Obr. 7.24 Současné uspořádání layoutu výrobní haly	147
Obr. 7.25 Potřeba pracovníků k zajištění provozu současného systému manipulace	148
Obr. 7.26 Návrh alternativního uspořádání layoutu výrobní haly	149
Obr. 7.27 Technologie pro provoz inteligentní tažné soupravy	150
Obr. 7.28 Návrh okruhů systému manipulace Milkrun 1	150

Obr. 7.29 Návrh okruhů systému manipulace Milkrun 2	151
Obr. 7.30 Potřeba manipulačních platforem – vozík.....	153
Obr. 7.31 Potřeba manipulačních platforem – podvozek	153
Obr. 7.32 Počet obslužených požadavků během jedné jízdy – Milkrun 1	154
Obr. 7.33 Počet obslužených požadavků během jedné jízdy – Milkrun 2	154
Obr. 7.34 Zpoždění zahájení výroby zakázky	155
Obr. 7.35 Vymezení zón pro stanovení potřebné velikosti logistických ploch.....	156
Obr. 7.36 Potřeba místa – Zóna 1	156
Obr. 7.37 Potřeba místa – Zóna 2	157
Obr. 7.38 Potřeba místa – Zóna 3	157
Obr. 7.39 Potřeba místa – Zóna 4	158
Obr. 7.40 Potřeba pracovníků k zajištění provozu systému manipulace Milkrun.....	158
Obr. 7.41 Rozložení poptávky v čase	160
Obr. 7.42 Layout skladu a rozdělení vychystávacích pozic dle obrátkovosti	163
Obr. 7.43 Umístění procesů a tok vychystaných objednávek v layoutu skladu	164
Obr. 7.44 Výkonový profil vychystávání objednávky	166
Obr. 7.45 Layout skladu po provedení reorganizace.....	167
Obr. 7.46 Detail pracoviště balení a kontroly, umístění spádového regálu.....	168
Obr. 7.47 Výkonový profil vychystávání objednávky – návrh nového systému	171

SEZNAM TABULEK

Tabulka 4.1 Aplikace SCOR modelu v modelování reálných dodavatelských systémů.....	49
Tabulka 4.2 Empirické rozdělení četností a pravděpodobností.....	59
Tabulka 4.3 Aplikace DES v modelování reálných systémů	61
Tabulka 4.4 Aplikace SD v modelování reálných systémů.....	64
Tabulka 4.5 Aplikace ostatních simulačních metod v modelování reálných systémů	66
Tabulka 4.6 Literární řešerše využité při stanovení významu simulačních metod v modelování dodavatelských systémů	67
Tabulka 5.1 Kritéria metodiky výběru simulačního softwaru.....	72
Tabulka 5.2 Aplikace simulačního nástroje Arena v modelování dodavatelských systémů	76
Tabulka 5.3 Aplikace simulačního nástroje Simul8 v modelování dodavatelských systémů	78
Tabulka 5.4 Aplikace simulačního nástroje Witness v modelování dodavatelských systémů	84
Tabulka 6.1 Pravidlo pro pojmenování elementů v obecném simulačním modelu materiálových toků vytvořeném v prostředí Witness	92
Tabulka 7.1 Úspory celkových nákladů na distribuci	138
Tabulka 7.2 Charakteristiky distribučních kanálů na základě prodejů v naturálních jednotkách a počtů realizovaných objednávek	159
Tabulka 7.3 Zvyklosti zákazníků z pohledu rozložení objednávek během týdne	160
Tabulka 7.4 Počet položek v jedné objednávce.....	161
Tabulka 7.5 Objednaná množství položky v kusech	161
Tabulka 7.6 Rozklad procesů tvořících výkonový profil na činnosti.....	166
Tabulka 7.7 Simulované scénáře	170
Tabulka 7.8 Výstupy simulace	170

SEZNAM POUŽITÝCH ZKRATEK

CPFR	Collaborative planning, forecasting and replenishment
CVRP	Capacitated vehicle routing problem
CWS	Clarke&Wright's Savings algoritmus
DES	Discrete event simulace
ECR	Efficient customer response
EDI	Electronic data interchange
FIFO	First in, first out
GPS	Global Position System
LA	Lokalizační algoritmus
LIFO	Last in, first out
PSMS	Algoritmus simulace pohybů skladové zásoby
QR	Quick response
SC	Supply chain
SCOR model	Supply chain operations reference model
SD	System dynamics
SCM	Supply chain management
SSCM	Sustainable supply chain management
VB	Visual basic
VL	Visual logic
VMI	Vendor managed inventory
VNA	Very narrow aisle
VRP	Vehicle routing problem
WMS	Warehouse management systém
ZMS	Základní modelová struktura

REJSTŘÍK

A

Aarts – van Laarhovenova rovnice · 39
adaptační algoritmus · 28
adaptivní ochlazování · 38
agent based simulace · 65
agregovaná preferenční účelová funkce · 16
algoritmus lokální optimalizace · 37
algoritmus simulovaného žíhání · 25
amplification effect · 5
analytical hierarchy process · 41
analytical network process · 41
analytické modely · 41
ant colony · 25, 35, 36
antagonistická hra · 42
Anylogic · 74
Arena · 75
attribute · 82
AutoMod · 74

B

Besselův elipsoid · 122
binární programování · 15
bipartitní graf s ohodnocením · 25
buffer · 79
bullwhip effect · 5

C

case based reasoning · 31
celočíselné programování · 15
cirkulace materiálových toků · 45
Clarke-Wright algoritmus · 128
closed-loop · 9
collaborative planning, forecasting and replenishment · 11
cradle to cradle · 45
cradle to grave · 45

D

data envelopment analysis · 41
defuzzyfikace · 31
descent algoritmus · 36
diagram kauzálních smyček · 63
disciplína fronty · 79
discrete event simulace · 60

distribution · 83
dodavatelský řetězec · 8
dodavatelský systém · 8
dopravní simulace · 65
down-hill · 36
dynamická simulace · 5, 49
dynamické programování · 14, 17

E

efekt biče · 5
efficient customer response · 10
electronic data interchange · 10
Emplant · 74
e-rám · 149
Erlangovo rozdělení · 22
exponenciální ochlazování · 38
exponenciální rozdělení · 22

F

fenotyp · 33
Fibonacciho generátor · 58
fill rate · 144
first in, first out · 143
fitness funkce · 33
Flexsim · 74
Forresterův efekt · 5
function · 83, 91, 92, 110
fuzziifikace · 31
fuzzy logic · 25, 31

G

generátor náhodných čísel · 57
genetický algoritmus · 33
genotyp · 33
gitterbox · 150
global position system · 122
globální účelová funkce · 16

H

Hebbovo učení · 30
horizontální integrace · 9
hra s konstantním součtem výher · 42
hra s nekonstantním součtem výher · 42
hra s nulovým součtem výher · 42

hra v normálním tvaru · 42
hybridní simulace · 65

I

indiferentní hráč · 42
input rule · 97, 107, 109, 117
inteligentní simulace · 65
inteligentní tažná souprava · 149
intenzita obsluhy · 23
intenzita provozu · 23
intenzita vstupu požadavků · 22
inventarizační analýza · 46
inverzní transformace · 58
iterativní učení · 29

J

jednomaticová hra · 42

K

kanban · 147
KLT box · 146
konečné hry · 42
kooperativní hra · 42
křížení · 34

L

labor rule · 98, 100, 102, 104
last in first out · 22
Lehmerova metoda · 57
lexikografická metoda · 16
life cycle analysis · 41
lineární dodavatelský systém · 45
lineární kongruentní generátor · 58
lineární ochlazování · 38
lineární programování · 14
lineární vícekritériální programování · 16
logistický řetězec · 8
logistický systém · 8
lokalizační algoritmus · 122

M

machine · 79
metoda ořezávání · 34

metoda větví a mezí · 15
metodika výběru simulačního softwaru · 71
milkrun · 150
modul Experimenter · 83
Monte Carlo simulace · 65
mutace · 35

N

nákladovost tržeb · 15
nekonečné hry · 42
nekooperativní hra · 42
nelineární programování · 15
neuronová síť · 28
NP-úplný problém · 36

O

obecný simulační model materiálových toků · 85
output rule · 98, 101, 103, 105, 107, 109, 117

P

part · 79
Petriho síť · 25, 26
piechart · 83
P-lower aproximace · 32
Poissonovo rozdělení · 22
posilující smyčky · 63
postup tvorby simulačního modelu · 54
princip antagonismu · 41
princip spolupráce · 10
problém obchodního cestujícího · 128
programovací jazyk · 60
Promodel · 74
P-upper aproximace · 33

Q

quick response · 10

R

racionální hráč · 41
reengineering procesů · 48
rentabilita tržeb · 15
reverzní logistika · 9
rough set · 32

rovnorné rozdělení · 57
rovnovážné smyčky · 63

Ř

řešení v čistých strategiích · 42
řešení ve smíšených strategiích · 42

S

SCOR model · 47
selekce · 34
simplexová metoda · 14
Simul8 · 76
simulace pohybů skladové zásoby · 125
simulační hodiny · 60
simulační hry · 65
simulační software · 71
smíšené programování · 15
software pro discrete event simulaci · 74
spreadsheet simulace · 49
stacionární stav obslužného systému · 23
standardizovaný profil výrobku · 47
strategie ochlazování · 38
Supply chain council · 49
supply chain management · 9
sustainable supply chain management · 9
system dynamics · 62

T

tabu search · 25, 40, 41
tabu seznam · 39
teoretický model biologického neuronu · 27
teorie her · 41
timeseries · 83
triple-bottom-line · 9
trojúhelníkové rozdělení · 58

U

učení bez učitele · 28
učení s učitelem · 28
umělá inteligence · 24

V

variable · 82
vážená ruleta · 34
vehicle routing problem · 128
vendor managed inventory · 10
very narrow aisle · 142
vícekriteriální programování · 16
vícekriteriální rozhodování · 41
virtuální simulace · 65
Visual basic · 60
Visual logic · 60
vylučovací metoda · 58

W

warehouse management systém · 142
Widrowovo učení · 30
Witness · 78
Witness Optimizer · 83

Z

základní modelová struktura · 89

Ž

životní cyklus výrobku · 45

INFORMACE O AUTOROVÍ

Jakub Dyntar je absolventem oboru Ekonomika a řízení chemických a potravinářských podniků na Vysoké škole chemicko-technologické v Praze, kde působí v Ústavu ekonomiky a managementu jako odborný asistent. Za svou doktorskou dizertační práci Řízení materiálových toků výrobků se sporadickou poptávkou pomocí dynamické simulace obdržel cenu Unipetrolu. Je autorem několika desítek vědeckých a odborných publikací zaměřených na oblast logistiky, operačního výzkumu a podnikové ekonomiky. V minulosti působil jako konzultant ve společnostech zabývajících se logistickým poradenstvím. Od roku 2007 provozuje společnost Jakub Dyntar, logistické poradenství.

Návrh a optimalizace dodavatelských systémů s využitím dynamické simulace

Autor: Jakub Dyntar

Vydavatel: FinEco
Třebešovská 14/2038
193 00, Praha 9
www.fin-eco.eu

Editor: Doc. Ing. Božena Kadeřábková, CSc.

Recenzenti: Prof. Ing. Gabriel Fedorko, Ph.D., ÚLPaD F BERG TU v Košiciach
Ing. Petr Jirsák, Ph.D., VŠE Praha, FPH, Katedra logistiky

Tisk: Česká technika – Nakladatelství ČVUT
Žitná 4
166 36 Praha

Vydání: první

Náklad: 200 ks

Praha, 2018

ISBN 978-80-86590-15-8